DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN
UNIVERSIDAD POLITÉCNICA DE VALENCIA

P.O. Box: 22012 E-46071 Valencia (SPAIN)


# Informe Técnico / Technical Report

| | | | |
|---|---|---|---|
| **Ref. No.:** | DSIC-II/04/10 | **Pages:** | 72 |
| **Title:** | GORMAS: A methodological guideline for organizational-oriented open MAS | | |
| **Author(s):** | S. Esparcia, E. Argente, V. Julián, V. Botti | | |
| **Date:** | May 18, 2010 | | |
| **Keywords:** | Multi-agent systems, Organizations, Open Systems | | |

V$^o$ B$^o$
Leader of research Group Author(s)

# 1 Introduction

GORMAS (**G**uidelines for **OR**ganizational **M**ulti-**A**gent **S**ystems) defines a set of activities for the analysis and design of Virtual Organizations, including the design of their organizational structure and their dynamics. With this method, all services offered and required by the Virtual Organization are clearly defined, as well as its internal structure and the norms that govern its behavior.

GORMAS is based on a specific method for designing human organizations, which consists of diverse phases for analysis and design. These phases have been appropriately adapted to the MAS field, this way to catch all the requirements of the design of an organization from the agents' perspective. Thus, the methodological guidelines proposed in GORMAS cover the typical requirement analysis, architectural and detailed designs of many relevant *Organization-Centered Multi-Agent Systems* (OCMAS) [1], (such as PASSI [2], SODA [3], AGR [4] and INGENIAS [5]) methodologies, but it also includes a deeper analysis of the system as an open organization that provides and offers services to its environment.

The proposed guideline allows being integrated into a development process of complete software, which may include the phases of analysis, design, implementation, installation and maintenance of MAS. GORMAS methodology is focused on the analysis and design processes, which are split into: mission and service analysis steps (analysis phase); and organizational and organization dynamics design steps (design phase). Implementation is carried out in the THOMAS [6] framework which mostly covers the organization software components that are required, such as organizational unit life-cycle management, service searching and composition and norm management.

This methodology is specified in order to design large scale, open and service-oriented MAS, where organizations are able to accept external agents into them. In order to model this kind of systems, GORMAS is supported by a CASE tool named EMFGormas [7], that uses the MDA Eclipse Technology. This technology requires defining a platform independent unified meta-model that describes the modeling language in a formal way, establishing the primitives and syntactic-semantic properties of organizations and multi-agent system. The tool offers several graphical editors, one for each view of the model, but diagrams are stored in a unique model.

GORMAS is composed of four phases (see fig. 1), covering the analysis and design of a MAS: first activity is **mission analysis**, a phase that implies the analysis of the system requirements, the use cases, the stakeholders and the global goals of the system; the **service analysis** phase specifies the services offered by the organization to its clients, as well as its behavior, and the relationships between these services; the **organizational design** step defines the structure for the Virtual Organization, establishing the relationships and restrictions that exist in the system; and finally, at the **organization dynamics design**

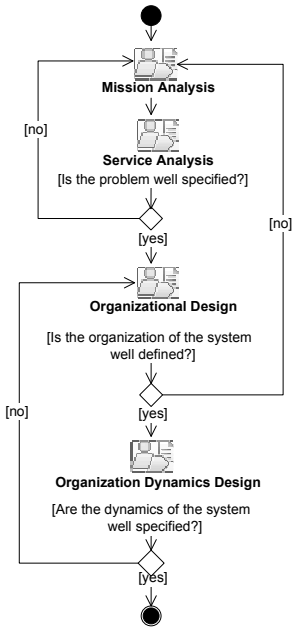



Figure 1: GORMAS design process

step, communicative processes between agents are established, as well as processes that control the acquisition of roles along with processes that enable controlling the flow of agents entering and leaving the organization. Additionally, some mechanisms (norms) that are used to control the system are defined. Finally, the organization dynamics design step is responsible of designing guides that establish a suitable reward system for the organization.

GORMAS metamodel takes into account the main aspects of a Virtual Organization: (i) the *Structural Dimension*, describing the components of the system and their relationships; (ii) the *Functional Dimension*, detailing the specific functionality of the system, based on services, tasks and goals, as well as system interactions, activated by means of goals or service usage; (iii) the *Dynamical Dimension*, which details the dynamic enactment of roles by agents, their dynamic participation into units and their interactions; (iv) the *Environmental Dimension*, describing the environment in terms of its resources and how agents can perceive and act on them; (v) and the *Normative Dimension*, defining the organizational norms and normative goals that agents must follow, including sanctions and rewards. All of these five dimensions are represented at the GORMAS metamodel.

In the following, GORMAS methodology will be described using the SPEM 2.0 notation, by initially considering its whole process and its metamodel, and then its four phases, describing the process roles, the activity details and the work products involved on each phase.

Some interesting GORMAS references are:

- E. Argente. *GORMAS: Guias para el desarrollo de Sistemas multi-agente abiertos basados en organizaciones.* PhD thesis, Universidad Politecnica de Valencia, 2008.

- E. Argente, V. Julian, and V. Botti. *MAS Modelling based on Organizations.* In Proc. Agent Oriented Software Engineering 2008, pages 1-12, 2008.

- E. Argente, V. Botti and V. Julian. *GORMAS: An Organizational-Oriented Methodological Guideline for Open MAS.* In Proc. Agent Oriented Software Engineering 2009 pages 85-96, 2009.

## 1.1 The GORMAS process life-cycle

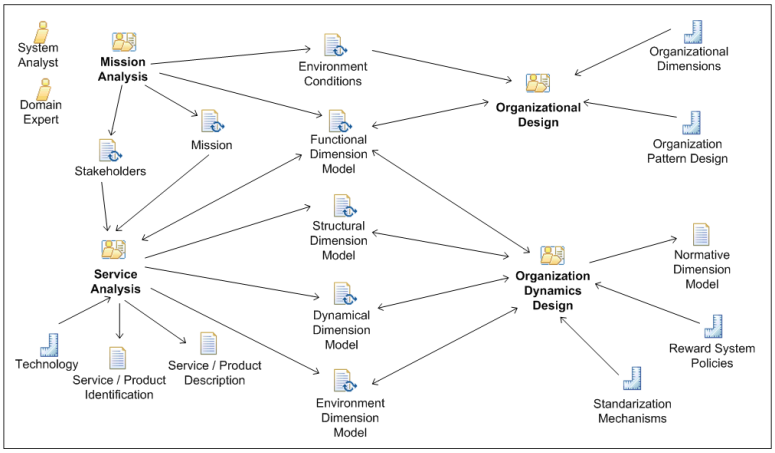



Figure 2: Detailed GORMAS design process

GORMAS follows a four-phase, iterative life cycle. At figure 2, all phases are shown, along with the documents and models generated by the methodology:

- **Mission Analysis**: describes the global goals of the system, the services and products that the system provides to other entities, the stakeholders and the conditions of the environment.

- **Service Analysis**: describes the type of products and services, and the tasks and goals related to its production; the resources and applications needed for offering the system functionality and the roles related with the stakeholders.

- **Organizational Design**: organizational dimensions are defined and they are employed to define a suitable structure for the organization.

- **Organization Dynamics Design**: communication between agents, processes concerning roles and agents, mechanisms for the control of the system, like norms, and a reward system, are defined.

On every phase, one or some of the diagrams that represent the five dimensions of the metamodel (that will be described on the next section) are updated. Additionally, there are some documents for describing the environment conditions, system mission, stakeholders, services and products, that are defined on the first two phases of the methodology and are lately employed through the whole process.

## 1.2 GORMAS metamodel

This section describes the GORMAS metamodel, which is made up of five models: functional, structural, dynamical, environment and normative dimensions. Figure 3 shows a simplified version of the GORMAS metamodel, representing its main entities and relationships.

In this figure, it is possible to identify the five dimensions that compose the GORMAS metamodel. The most representative entities and relationships from the metamodel are shown. They have been selected in order to provide a quick overview of the GORMAS metamodel entities and the relevant relationships for every dimension. These elements are represented around the *Organizational Unit* entity. Surrounding it, the five dimensions that compose the metamodel are shown by using different colors (see caption box on figure 3).

The *Functional Dimension* details the functionality of the system based on services, tasks and goals. This simplified metamodel figure depicts the relationships that *Organizational Unit* maintains with *objectives* and *services*. It represents the *products* and *services* offered or required by the *stakeholders*.

The *Structural Dimension* specifies the elements of an *Organizational Unit*, such as *roles*, *agents* and *norms*. In this simplified view, the *'contains'* relationship between *roles* and *Organizational Units* has been depicted.

GORMAS describes the environment of a system by means of the resources and applications (called the environment elements) that are surrounding the organization. Therefore, the *Environment Dimension* defines the resources and applications that the organization will have available. Additionally, *ports* are entities that manage the access to services and products from the environment.

The *Dynamical Dimension* defines the interactions between agents, as well as the mental states of the system. Finally, the *Normative Dimension* is responsible of describing the *norms* that will manage the behavior of the entities of the organization.

Organizations are structured by means of Organizational Units (OU), which represent a set of agents that carry out some specific and differentiated activities or tasks, following

a predefined pattern of cooperation and communication. An OU is formed by different entities along its life cycle which can be both single agents or other organizational units, viewed as a single entity. System entities are capable of offering and/or requesting services and their behavior is motivated by their pursued goals. Services represent the functionality that agents offer to other entities, independently of the concrete agent that makes use of it. Moreover, an organizational unit can also publish its requirements of services, so then external agents can decide whether participate inside, providing these services.

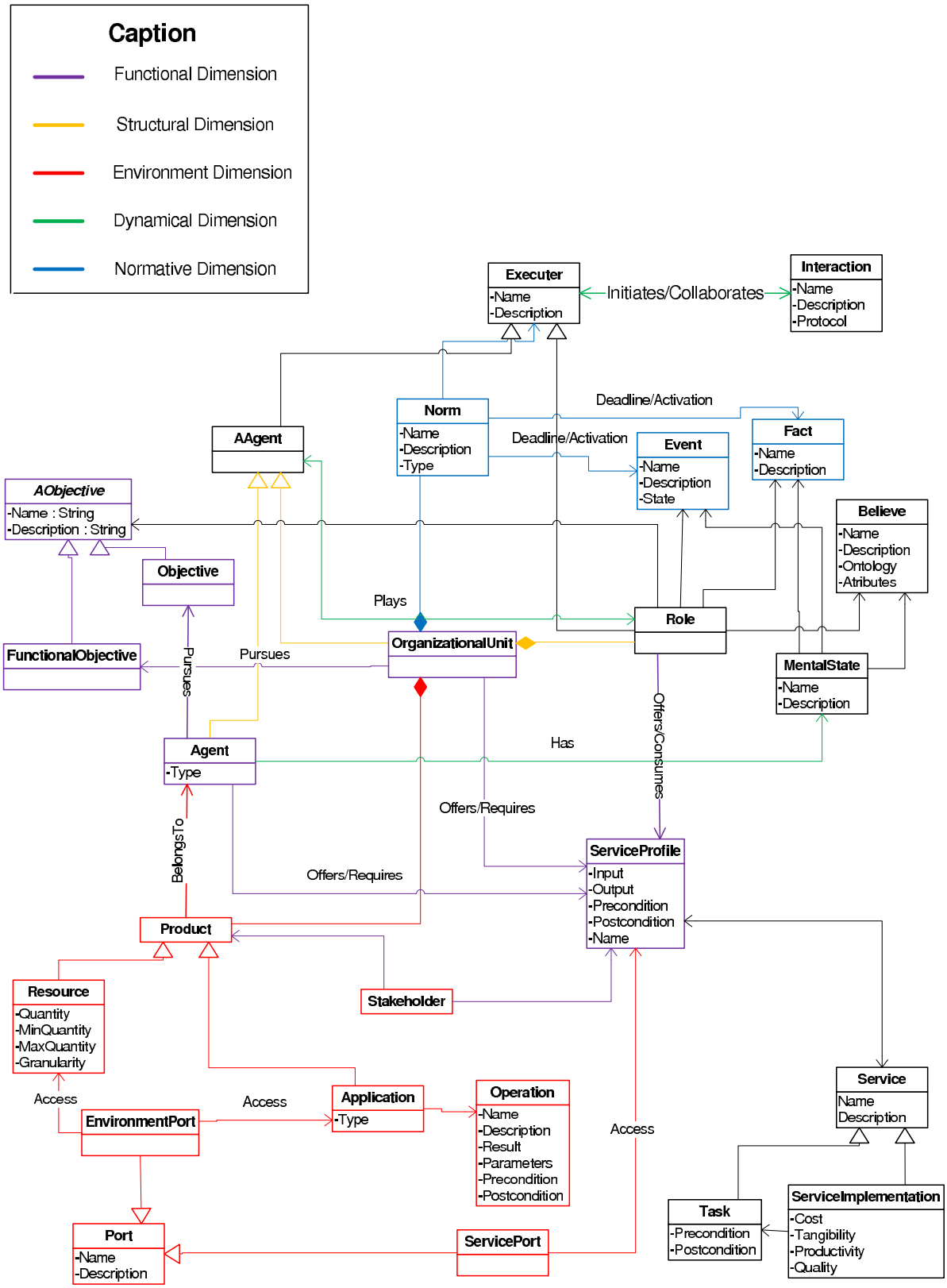



Figure 3: Unified and simplified metamodel for GORMAS

### 1.2.1 Definition of MAS metamodel elements

| **Concept** | **Definition** | **Domain** |
|---|---|---|
| Organizational Unit (OU) | A set of agents that carry out some specific and differentiated activities or tasks by following a predefined pattern of cooperation and communication. An OU is formed by different entities along its life cycle which can be both single agents or other organizational units, viewed as a single entity. | Structural, functional, dynamical and environment dimensions |
| AAgent | An entity of the system, which represents an atomic entity or a group of members of the organization, seen as unique entity from outside | Structural and dynamical dimensions |
| Agent | An entity capable of perceiving and acting into an environment, communicating with other agents, providing and requesting services/resources and playing several roles. | Structural, functional, dynamical and environment dimensions |
| Product | An entity that is contained into an OU or that belongs to an specific agent. It can be an application or a resource. | Structural and functional dimensions |
| Resource | It is an environment object that does not provide a specific functionality, but is essential for task execution. | Structural and environment dimension |
| Application | It is a functional interface that does not satisfy a rational criteria | Structural and environment dimensions |
| Role | An entity representing a set of goals and obligations, defining the services that an Agent or an OU could provide and consume. | Structural, functional and environment dimensions |
| AObjective | It is a goal pursued by a role. An AObjective could be a functional objective or an operational goal. | Functional dimension |
| Functional Objective | A functional objective is a non-functional requirement (softgoals) that could be defined to describe the global behavior of the organization. | Functional dimension |
| Objective | An objective is a specific goal that agents or roles have to fulfill. It can be refined into specific objectives. | Functional dimension |
| Service Profile | It is the description of a service that the agent might offer to other entities | Functional dimension |

| | | |
|---|---|---|
| Service Implementation | It is a service specific functionality which describes a concrete implementation of a service profile | Functional dimension |
| Service | An entity describing the service functionality and represents concrete tasks, task-flows or service composition | Functional dimension |
| Task | An entity that represents a basic functionality, that consumes resources and produces changes in the agent's Mental State. | Functional and dynamical dimensions |
| Norm | It is a coordination mechanism and represents a specific regulation for the system. | Structural and normative dimensions |
| Stakeholder | It is a group that the organization is oriented to and interacts with the OUs. | Functional dimension |
| Mental State | It is a set of believes, events and facts that define the current state of an agent. | Dynamical dimension |
| Believe | It is something that an agent (or a role taken by an agent) thinks that it is true or will happen. | Dynamical dimension |
| Fact | It is something that is true at the system's domain. | Dynamical and normative dimensions |
| Event | It is something that changes the state of the system when it occurs. | Dynamical and normative dimensions |
| Interaction | An entity defining an interaction between agents. | Dynamical dimension |
| Interaction Unit | A performative employed during the interaction. | Dynamical dimension |
| Condition | An entity that allows defining the sequence of tasks depending on a condition. | Dynamical dimension |
| Executer | A participant in an interaction. It can be an AAgent or a Role. | Dynamical, environment and normative dimensions |
| Port | A point managed by the Executer, used to access a service or an environment element | Environment dimension |
| Service Port | A service publication point that offers the chance of registering and searching for services by their profile. | Environment dimension |
| Environment Port | An entity that describes who can use each resource or application and who is responsible of giving these permissions. | Environment dimension |
| Operation | An entity containing the attributes of an Application entity: name, description, results, parameters, preconditions and postconditions. | Environment dimension |
| Split Generalization | An entity that represents that a Task or a Service can be split. | Structural dimension |

### 1.2.2 Detailed description of the GORMAS metamodel dimensions

The **Structural Dimension** (Figure 4) describes the components of the system and their relationships. It allows defining the static components of the organization, i.e. all elements that are independent of the final executing entities. More specifically, it defines:

- the entities of the system (*AAgent*), which represent an atomic entity (*Agent*) or a group of members of the organization (*Organizational Unit*), seen as a unique entity from outside.
- the *Organizational Units* (OUs) of the system, that can also include other units in a recursive way, as well as single agents.
- the *Roles* defined inside the OUs. In the *contains* relationship, a minimum and maximum quantity of entities that can acquire this role can be specified. For each role, the *Accessibility* attribute indicates whether a role can be adopted by an entity on demand (external) or it is always predefined by design (internal). The *Visibility* attribute indicates whether entities can obtain information from this role on demand, from outside the organizational unit (public role) or from inside, once they are already members of this organizational unit (i.e. private role). A hierarchy of roles can also be defined with the *InheritanceOf* relationship.
- the organization social relationships (*SocialRelationships*). The type of a social relationship between two entities is related with their position in the structure of the organization (i.e. information, monitoring, supervision), but other types are also possible. Moreover, a condition on when this social relationship is active can also be established.
- the *products* (resources/applications) available by an OU.
- the *norms* that control the global behavior of the members of the OU.
- All contains relationships include conditions for enabling a dynamical registration/deregistration of the elements of an OU through its lifetime.

The **Functional Dimension** (Figure 5) that details the specific functionality of the system, based on services, tasks and goals, as well as system interactions, activated by means of goals or service usage. It allows defining the functionality of the organizational units, roles and agents of the MAS. More specifically, it defines:

- the functionality of the Organizational Units:

  – the *functional objectives* that are pursued by the organization, i.e., non-functional requirements (softgoals) that can be defined for describing the global behavior of this organization.

  – the *stakeholders* that interact with the OU.

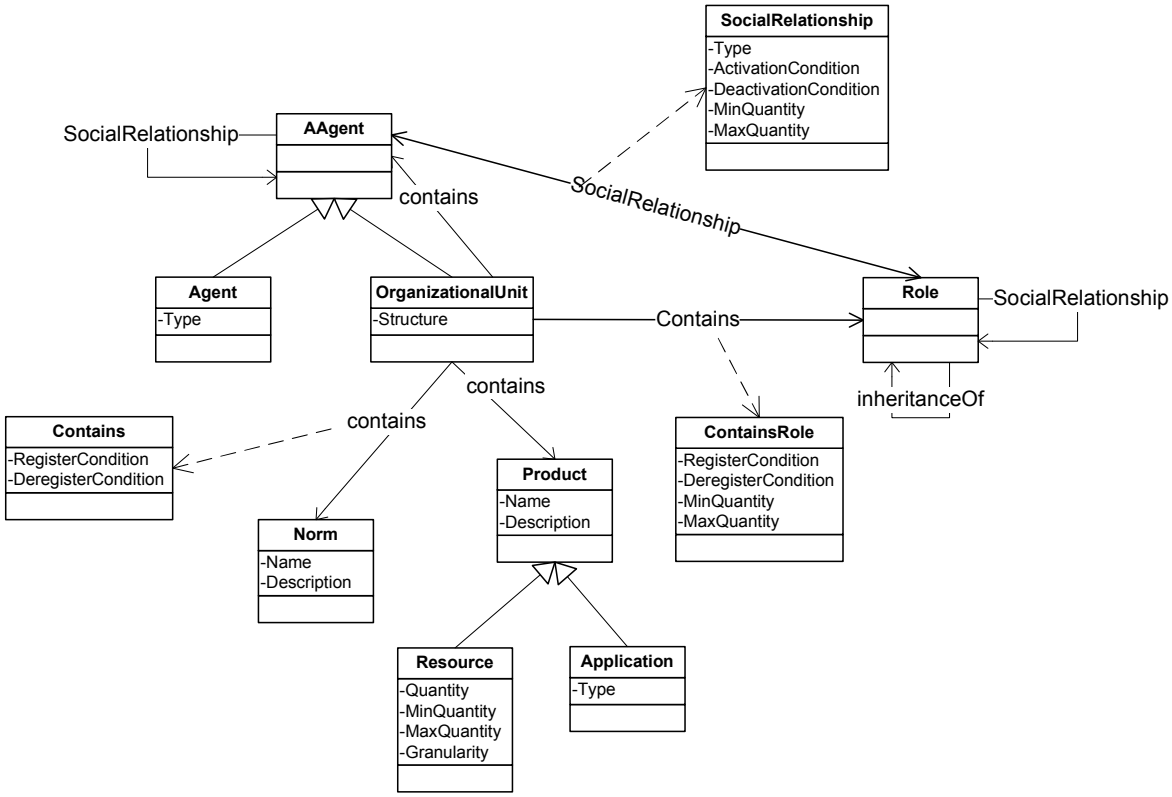



Figure 4: *Structural Dimension* diagram.

– the results that the organization offers (*products* and *services*, which are described using service profiles). In case of services, a specific implementation of the service can be defined in the *offers* relationship (*ServiceImplementation* attribute), which may be registered in a service directory (*RegisterPort* attribute), so then other entities can find it. Conditions for controlling this registration process can also be specified (*RegisterCondition* and *DeregisterCondition* attributes).

– the services that are required by the organization (*Require* relationship). This "requires" relationship is similar to job offer advertising of human organizations, in the sense that it represents a necessity of finding agents capable of providing these required services as members of the organization.

– the organization needs from its providers (*Consumes* relationship).

- the composition of goals:

– The *AObjective* components, which can be *functional objectives* (i.e. softgoal or non-functional requirements) or operational goals (i.e. hardgoals or *objective*).

– The *Functional objectives* represent the expected results of organizational units, which are split into the specific and measurable results that their members are expected to achieve.

–The *Objectives* represent the operational goals, i.e., the specific goals that agents or roles have to fulfill. They can also be refined into more specific objectives. They might be related with a Task or Interaction needed for satisfying this objective.

- the functionality of the Roles:

  – the *goals* (*AObjective*) pursued by a role, which can be *functional objectives* (i.e. softgoal or non-functional requirements) or operational goals (i.e. hardgoals or simply *objective*). In the *pursues* relationship, activation and deadline conditions can be defined to stablish a temporal timeline in which the objective is followed. In case of an operational goal (*objective*), a satisfaction or fail condition can be defined in order to stablish when this objective has been fulfilled, as well as a Task or Interaction that enables reaching this goal.

  – the services (*ServiceProfile*) related to the role, i.e., the services that the role is enabled to offer or provide to other entities.

  – the *tasks* that the role is responsible for, i.e. the specific functionality that the role is expected to be able to carry out.

- the functionality of the Agents:

  – the *objectives* pursued by agents. Activation and deadline conditions can be defined to stablish a temporal timeline in which the objective is followed. Moreover, a satisfaction or fail condition can be defined in order to stablish when this objective has been fulfilled.

  – the services (*ServiceProfile*) related to the agent, i.e., the services that the agent might offer to other entities. When adopting a role as a member of an organization, the concrete set of services that the agent will be allowed to provide is determined by its own set of offered services and those ones related to the adopted role.

  – the *tasks* that the agent is responsible for, i.e. the set of tasks that the agent is capable of carrying out.

- the composition of tasks and services:

  – The *Service* component describes the service functionality and represents both concrete tasks, task-flows or service composition (*Invokes* relationship). This *Service* component can be split into other Service components, thus allowing service refinement or task composition.

  – A *Task* represents a basic functionality, that consumes and produces changes in the agent's Mental States.

  – The *order* relationship between tasks, in which ordering conditions can be defined, as well as interactions. The entity *Condition* allows defining the sequence of tasks depending on a condition.

– The service interface (*ServiceProfile*), which indicates activation conditions of the service (preconditions), its input and output parameters and its effects over the environment (postconditions). It will be lately used in an OWL-S service description.

– The service specific functionality (*ServiceImplementation*), which describes a concrete implementation of a service profile.

– The service composition, by means of *invokations* between services.

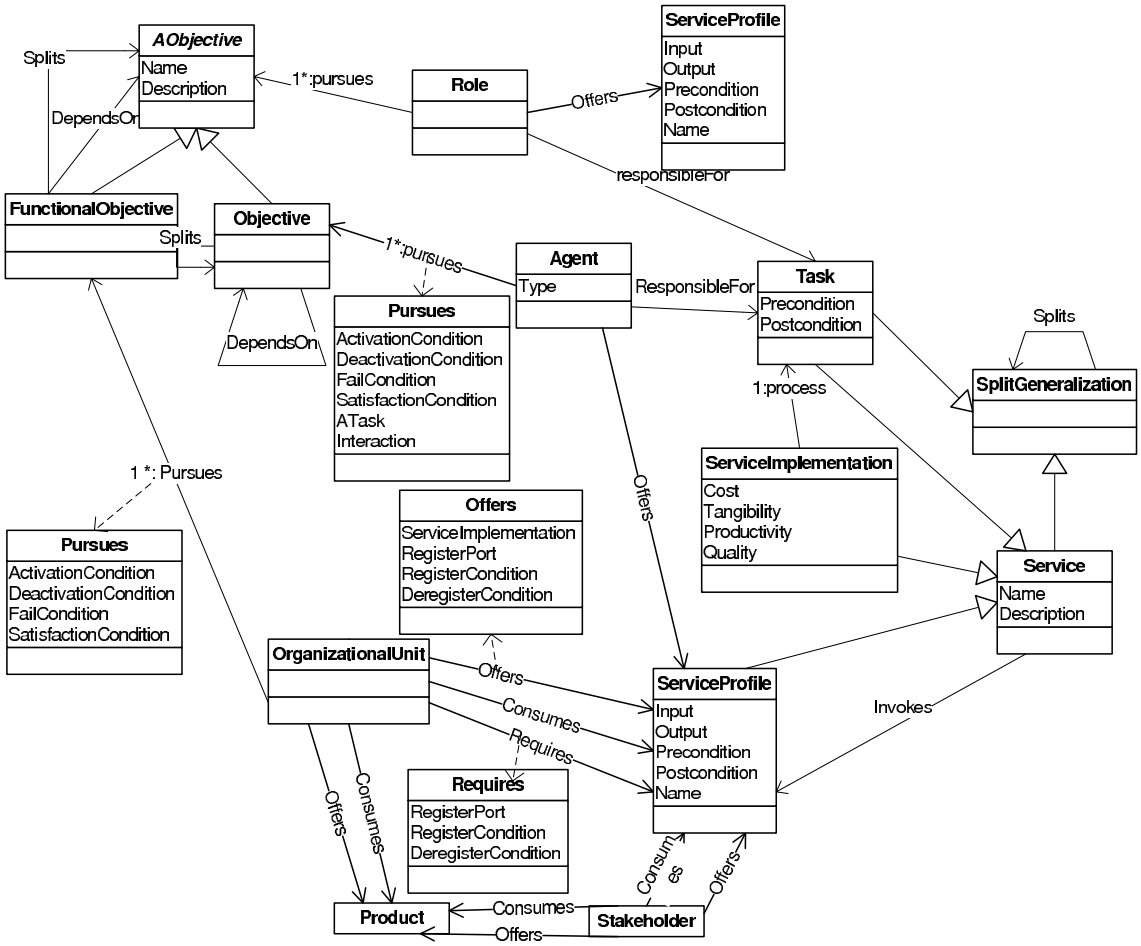



Figure 5: *Functional Dimension diagram.*

The **Dynamical Dimension** (Figures 6 and 7) defines the role enactment process, the interactions between agents, as well as the mental states of the entities of the system. More specifically, it defines:

- the *roles* that the organizational unit may play inside other organizational units (*Plays* relationship), when considered as a unique entity. *ActivationCondition* and *Leave-Condition* attributes of this relationship indicate in which situation an OU acquires or leaves a role.

- the *roles* played by each agent. *ActivationCondition* and *LeaveCondition* attributes of this *play* relationship indicate in which situation an agent can acquire or leave a role.
- the *Mental States* of the agent, using believes, events and facts.
- the sequence of interactions:

  – The participants of the interaction (*Executer*). The *Initiates* and *Collaborates* relationships indicate the sequence of activities (task and services) that have been executed in an interaction. The *Collaborates* relationship represents a response activity.

  – The performatives (*InteractionUnit*) employed during the interaction.

  – The entity *Condition* allows changing the sequence of interactions depending on a condition.

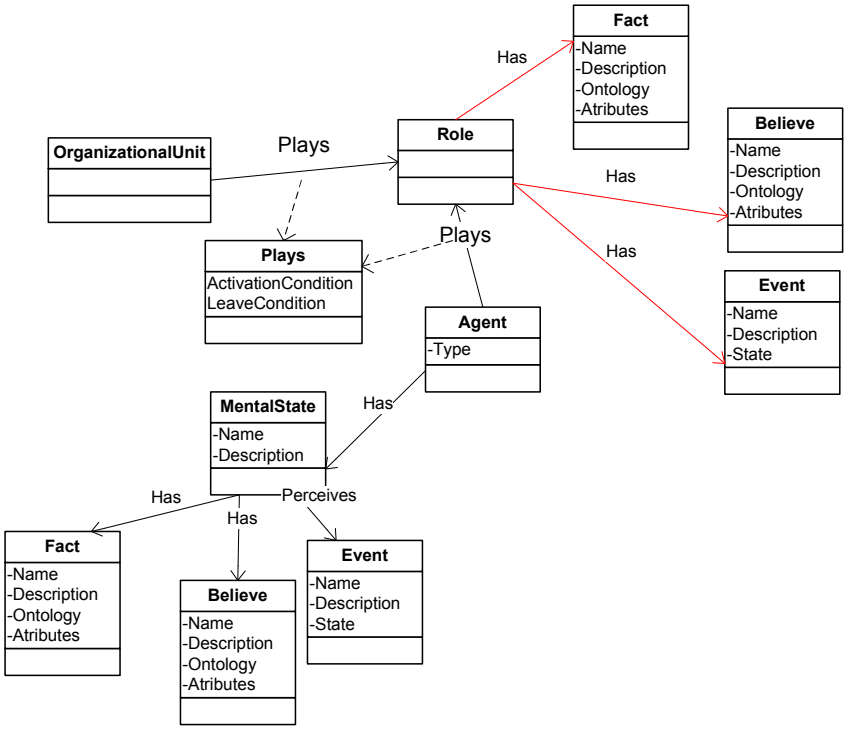



Figure 6: *Dynamical Dimension diagram (I).*

The **Environment Dimension** (Figure 8) describes the environment elements (resources and applications), along with the agents' behavior. It also allows defining service ports and the coordination mechanisms. More specifically, it defines:

- The *products* that are contained in the Organizational Units or that belong to specific agents. They can be applications or resources.

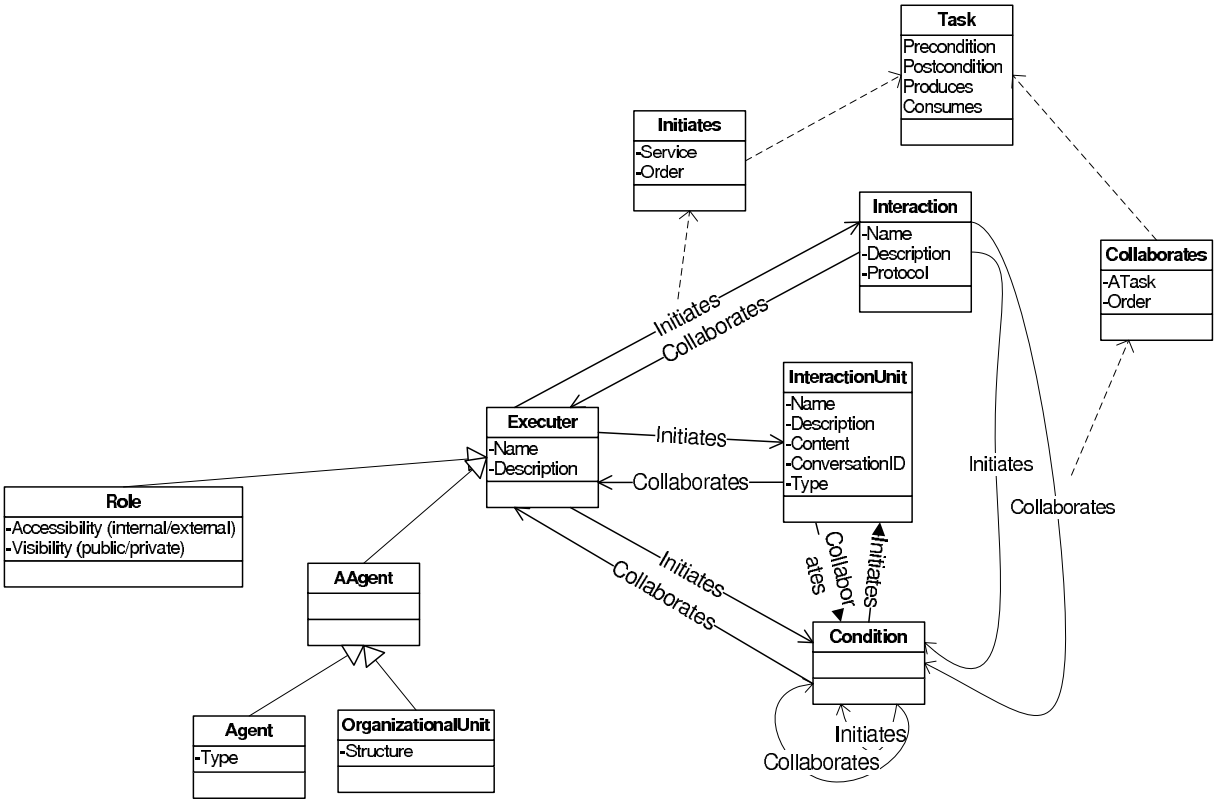



Figure 7: *Dynamical Dimension diagram (II).*

- *Resources* represent environment objects that do not provide a specific functionality, but are indispensable for task execution. They can be consumable or not, have an initial state (*Quantity*), a lower and upper threshold (*MinQuantity*, *MaxQuantity* attributes) and a *granularity* capacity.
- *Applications* represent functional interfaces that are described with a *name*, several *parameters*, *preconditions*, *postconditions* and *results*.
- Products can be associated to *environment ports*, thus describing who can use each resource or application and who is responsible of giving these permissions.
- A *service port* is considered as a service publication point that offers the possibility to register and search services by their profile
- The *Executer* that controls (*Manages* relationship) each port (environment/service port) defines the usage permissions over them.
- The *Executer* that is *allowed to* use the port can employ it for perceiving or acting over the associated product, in case of a environment port; or for registring new services or accesing existing ones, in case of a service port.

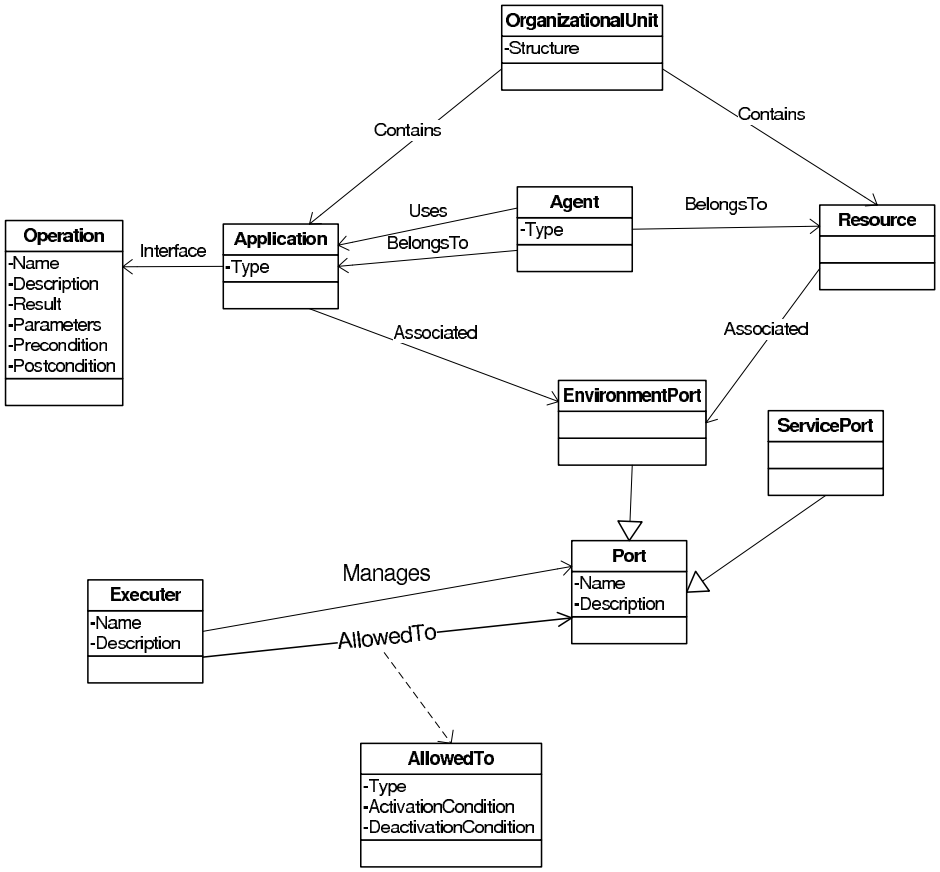



Figure 8: *Environment Dimension diagram.*

Finally, the **Normative Dimension** (Figure 9) describes normative restrictions over the behavior of the system entities, including organizational norms and normative goals that agents must follow, including sanctions and rewards. More specifically, it defines:

- the *Norm* concept, which represents a specific regulation. The properties of the norm detail all facts and events of the environment that provoke the activation or deactivation of the norm.
- The entity (*Executer*) to whom the norm is applied (*Concerns* relationship).
- The *Executer* that is responsible of: monitoring the norm satisfaction (*Controller* relationship); applying punishments (*Defender* relationship); and/or applying rewards (*Rewarderer* relationship).
- The *Service* attribute of all these relationships indicates which task or service will be invoked when monitoring this norm and when punishing or rewarding agents.

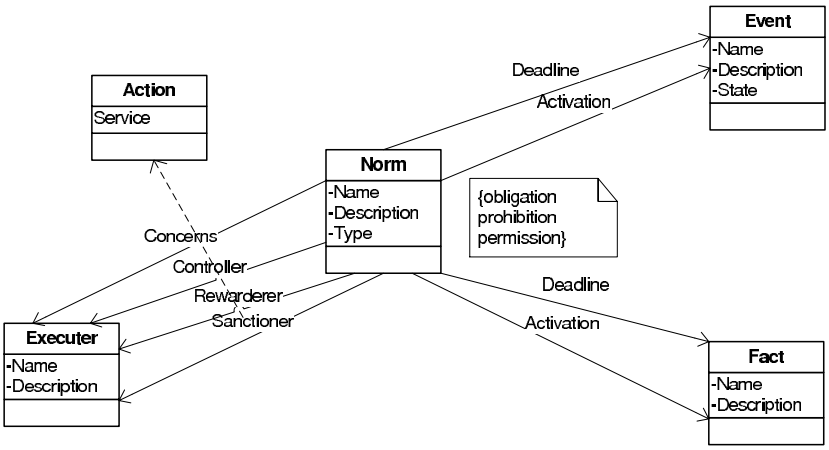



Figure 9: *Normative Dimension diagram.*

# 2 Phases of the GORMAS methodology

In this section, the phases that compose the GORMAS methodology will be described. GORMAS offers a set of guidelines to analyze the requirements of the system, to design the structure of the organization and to design the dynamics of an organization based on a multiagent system. This set of guidelines is aimed to define the services provided by an organization, its structure and the norms governing the organization, making it easy to analyze and design open MAS.

For each phase, a brief description of its scope will be given. Then, the process roles that are involved in this phase will be defined and activities and tasks from that phase will be described in detail. Finally, work products will be identified. Relationships between GORMAS metamodel elements and the work products will be identified.

At the following subsections, every step of the GORMAS process will be detailed.

## 2.1 Mission analysis

The first phase of the GORMAS methodology (figure 10) implies the analysis of the system requirements, identifying the use cases, the stakeholders and the global goals of the system. This step involves two different process roles, four work products (one model diagram and three text documents) and one guidance document, as described on figure 11. This phase is composed of five tasks (Identify Organization Results, Identify Stakeholders, Identify Environment Conditions, Define the System Mission and Justify the MAS System). The process flow at task level is reported in figure 10.

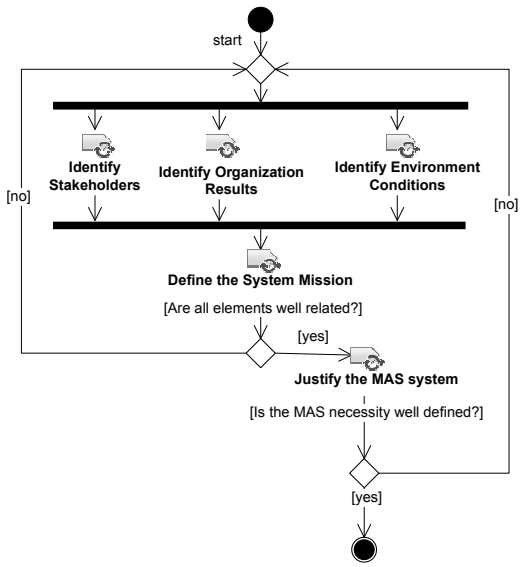



Figure 10: Activity diagram of *Mission Analysis* phase

As a result, a diagram of the *Functional Dimension Model* is drawn, detailing the products and services offered by the system, the global goals (mission) pursued, the stakeholders and the existing links between them, the system results as well as the resources or services needed.

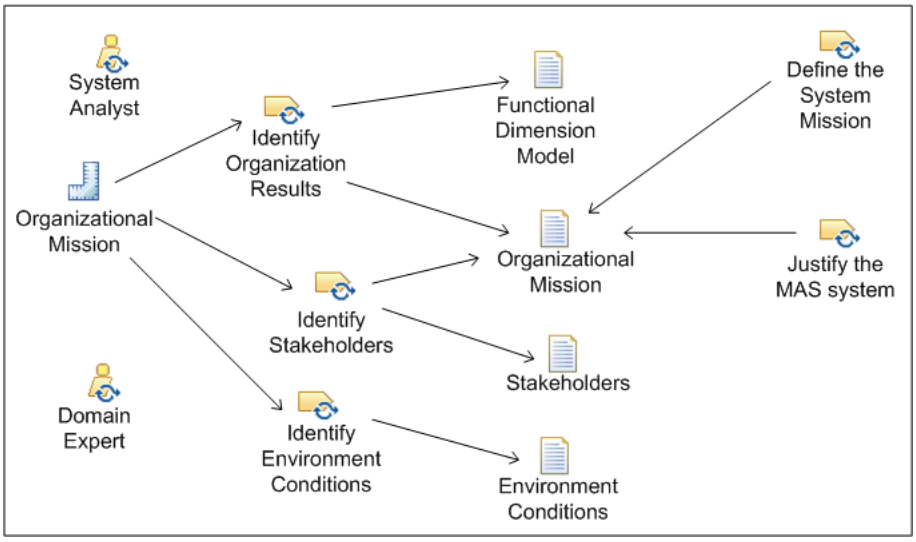



Figure 11: Resources and products used in *Mission Analysis* phase

### 2.1.1 Process roles

There are two roles involved in the *Mission Analysis* phase: the System Analyst and the Domain Expert. Both of them are described in the following subsections.

**System Analyst**

He is responsible for:

- Defining the mission and the context of the organization, by means of identifying the system results, the stakeholders and the environment of the organization.
- Creating the documents that define the mission of the system.
- Defining the *Functional Dimension Model* diagram.

**Domain Expert**

He is responsible of supporting the system analyst during the *Mission Analysis* phase, by giving him all the information that he could need.

### 2.1.2 Activity details

In the *Mission Analysis* step, it is defined:

- The global goals of the system (mission).
- The services and products that the system provides to other entities.
- The stakeholders with whom the system contacts (clients or suppliers of resources/services), describing their needs and requirements.
- The conditions of the environment or context in which the organization exists (i.e. complexity, diversity, restrictions, etc.).
- The justification of the existence of the MAS system that it is being designed, in order to look that the GORMAS definition on a MAS could contribute on defining an organization.

In order to identify all these items, five tasks are needed, detailed in table 1. These tasks are aimed to look for the system mission, by means of: (i) identifying the organization results; (ii) identifying the stakeholders and (iii) identifying the environment conditions. Moreover, global goals of the system are described. Finally, it is necessary to justify whether the GORMAS approach for creating organizations is suitable for the current problem under study.

| **Task** | **Task Description** | **Roles Involved** |
|---|---|---|
| Identify organization results | Describe the results (products or services) that the system provides, to understand what the result is, what it does and who is interested in. | System analyst and domain expert |
| Identify stakeholders | Identify and describe the main stakeholders that the organization is related to (external actors, clients, users, etc.) | System analyst and domain expert |
| Identify environment conditions | Identify and define the kind of environment in which the organization will be developed, knowing if it is a physical environment or a virtual environment; if it is a distributed environment, etc. | System analyst and domain expert |
| Define the System Mission | Identify the global goals pursued by the system. These goals compose the mission of the organization. | System analyst and domain expert |
| Justify the MAS system | Justify the existence of this kind of system, comparing it to other existing similar systems (that can use agents or not), and analyzing the advantages and disadvantages, and the singularities of the proposed system. | System analyst |

Table 1: *Mission Analysis* phase tasks

### 2.1.3 Work products

The following section describes the products generated on the *Mission Analysis* phase. Firstly, the *Functional Dimension Model* diagram is defined, and three documents, related to organizational mission, stakeholders and environment conditions are generated (see table 2).

Figure 12 describes their relation with the elements of the GORMAS metamodel. In this figure, each of the work products reports one or more elements from the GORMAS

meta-model; each MAS meta-model element is represented using an UML class icon and, in the documents, such elements can be Defined, reFined, Quoted, Related or Relationship Quoted as described below:

- defined: D label near the element symbol, this means that the element is introduced for the first time in the design of this work product (i.e. the MAS metamodel element is instantiated in this diagram).
- reFined: F label means that the MAS metamodel element is refined in the work product (for instance by means of attribute definition).
- related: R label, this means that an already defined element is related to another, or from a different point of view, that one of the MAS metamodel relationships is instantiated in the document.
- quoted: Q label, this means that the element has been already defined and it is reported in this work product only to complete its structure but no work has to be done on it.
- Relationship Quoted: RQ label means that the relationship is reported in the work product but it has been defined in another part of the process.

| **Name** | **Description** | **Work Product Kind** |
|---|---|---|
| Functional Dimension Model | A diagram using the GORMAS graphical notation (based on GOPPR notation) that details the specific functionality of the system, based on services, tasks and goals. | Behavioral |
| Organizational Mission | A document describing the basic aspects of the organization that will be defined. | Structured text |
| Stakeholders | A document describing the stakeholders that will take part on the organization. | Structured text |
| Environment conditions | A document describing the conditions that the environment of the organization will have. | Structured text |

Table 2: Products for *Mission Analysis* phase

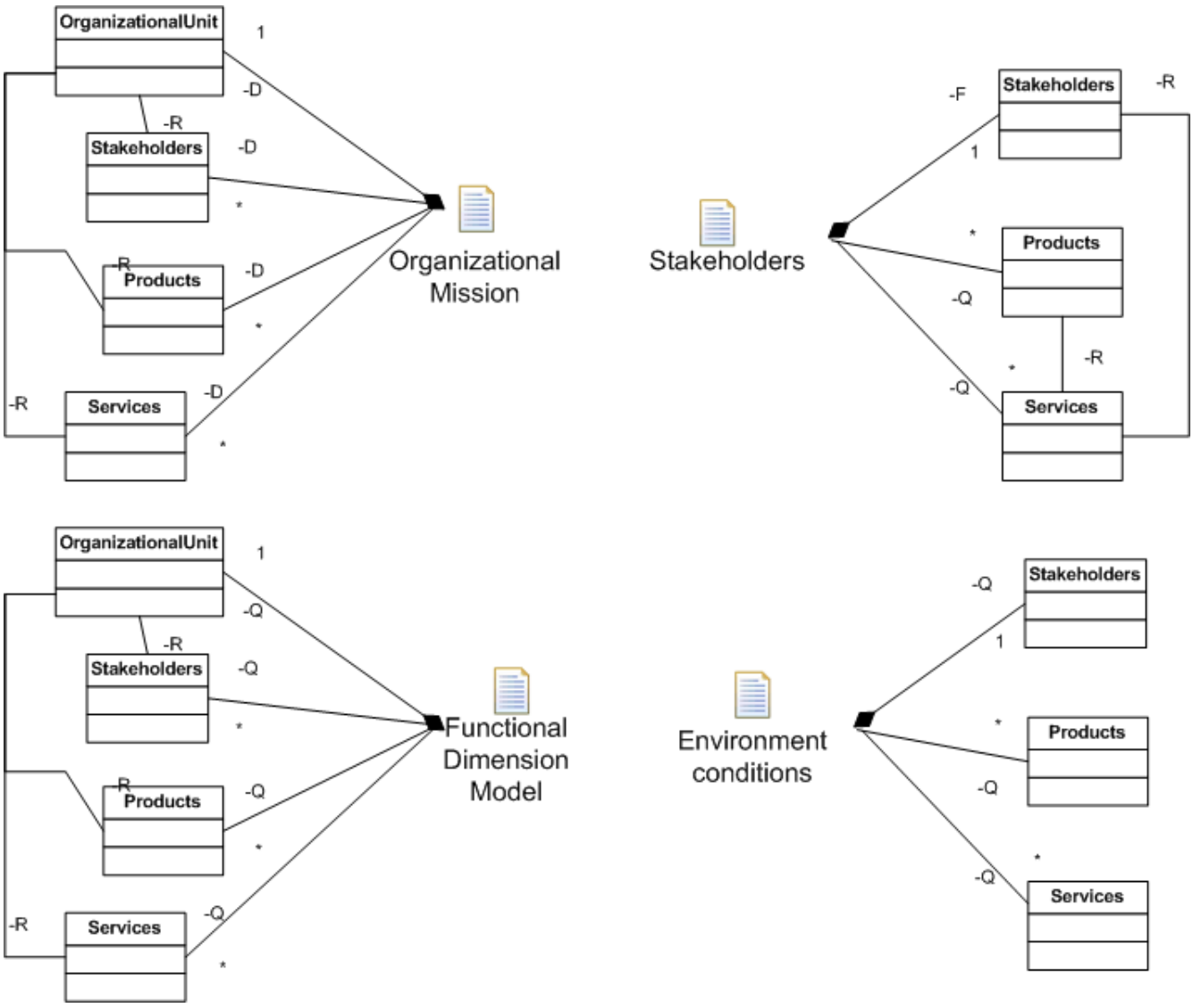



Figure 12: *Mission Analysis* phase. Relations between work products and metamodel elements.

**Organizational Mission**

This document is employed to define the mission of the organization that will be described. It is a structured text document. Its template is shown in table 3 and table 4 gives an example of this work product. As shown, it is necessary to give a name, a domain and an environment for the organization. Additionally, it is necessary to set the results that the system will provide with and the stakeholders that are interested on keeping a relationship with the organization. Finally, a justification for designing the system must be provided.

| **Organizational mission** |
|---|
| **Name:** general name of the system or organization to be generated |
| **Domain:** kind of market or interest area of the organization |
| **Results:** set of products or services offered by the organization to its clients<br>- *Purpose*: Description of the motivation by which this result is offered.<br>- Is it *tangible*?: If the result is storable, printable and/or reusable, it is a "product". If it is a used up functionality, it is a "service". |
| **Stakeholders:** actors that set up the market of the organization.<br>- Is it a *consumer*?: the actor consumes the products or services that the organization provides.<br>- Is it a *producer*?: the actor provides some resources or services that are required by the organization to work. |
| **Kind of environment:** location of the system (unique or distributed). Ability to access to the real and physical world. |
| **Context restrictions:** a set of restrictions that are imposed by the context or environment of the organization, and could affect to its structure, services, etc. |
| **Justification:** reason of the existence of the organization<br>- *Similar systems*: to detail the existing systems that provide a similar orientation than the one we are considering.<br>- *Advantages*: set of advantages that we want to afford with the new proposal. i.e., optimal use of the resources or services.<br>- *Disadvantages*: limitations that the new proposal has.<br>- *Singularities*: competitive elements of the organization. |

Table 3: Template for *Organizational Mission* document

| *Organizational Mission* |
|---|
| **Name**: Universidad Politecnica de Valencia (UPV) |
| **Domain**: Public university: education and research |
| **Results**:<br>• *Product:* **Requests** *Purpose:* A way to make any request to the system or outside it.<br>• *Product:* **Databases** *Purpose:* A product to save or extract relevant information about the organization.<br>• *Product:* **Bills** *Purpose:* Some documents that allow justifying the payments that the organization must afford.<br>• *Service:* **Budget management**. *Purpose:* A service to create, approve and modify the organization's budget.<br>• *Service:* **PhD management**. *Purpose:* A service to manage all the activities generated by the PhD degree studies.<br>• *Service:* **Staff management**. *Purpose:* This service manages all the activities related with the staff which works on the organization: contracts, salaries... |
| **Stakeholders**:<br>• Students: They need services related to their studies.<br>• Governing organs: They are responsible for the control of the organization.<br>• Teachers: They control teaching and researching related activities. |
| **Kind of environment:** The environment is distributed between the different departments of the organization. The system will be in touch with the real world. |
| **Context restrictions:** The organization must have a manager. |
| **Justification**: This is a system developed to manage the services that the University offers.<br>• *Similar systems*: Other universities or organizations divided into departments.<br>• *Advantages*: To facilitate the activities and procedures that must be performed by the system users.<br>• *Singularities*: The performance of the organization is better using an open system than using a closed system. |

Table 4: Example of *Organizational Mission* document

**Stakeholders**

This document is employed to describe the stakeholders of the organization, that have been defined in the *Organizational Mission* document. It is a structured text document. Its template is shown in table 5, table 6 gives an example of this work product. The identification of the stakeholders must be completed by providing the kind of stakeholder, the objectives that every group follows, their products and services provided and required, the benefits obtained by them and their position into the organization.

| **Stakeholders** | |
|---|---|
| **Name** | An identifier for the stakeholder |
| **Beneficiary** | Indicate whether the stakeholder is a primary (essential) or a secondary beneficiary. |
| **Type** | Indicate whether the stakeholder it is a client, a provider or a regulator. |
| **Objectives** | Describe the objectives pursued by the stakeholder. |
| **Requires** | A set of products and/or services that the stakeholder consumes. |
| **Provides** | A set of products and/or services that the stakeholder offers to the organization. |
| **Frequency** | To point out whether this stakeholder contacts with the organization frequently, occasionally or in an established period of time. |
| **Benefits** | Describe the benefits that the stakeholder wants to achieve. |
| **Decision power** | Indicate whether their needs are affecting to the requirements of products or services. |
| **Under the influence of the system?** | Indicate whether the organization can affect the interests of the stakeholder. |
| **Contribution** | To point out what the organization obtains from its relationship with the stakeholder. |

Table 5: *Stakeholders* document

| *Stakeholders* | | | |
|---|---|---|---|
| | **Stakeholder** | | |
| | **Students** | **Governing organs** | **Teachers** |
| *Beneficiary* | Primary | Primary | Primary |
| *Type* | Client | Provider | Client |
| *Objectives* | To manage the activities related with their studies. | To control the services and products provided by the organization. | To manage their educational and research activities. |
| *Requires* | | | |
| *Services* | PhD management | Budget management | Staff and PhD management |
| *Products* | Requests | Requests, databases, bills | Requests, databases, bills |
| *Provides* | | | |
| *Services* | | | |
| *Products* | Requests | Requests, bills | Requests, bills |
| *Participates* | | Staff management | |
| *Frequency* | Frequent | Frequent | Frequent |
| *Benefits* | To obtain a response about their needs. | To make sure that the organization is working correctly. | Financial help and research projects. |
| *Decision power* | No | Yes | Yes |
| *Under the influence of the system?* | Yes | No | Yes |
| *Contribution* | The organization will not be possible without them. | To validate resources of the organization. | Teaching, research... |

Table 6: Example of a *Stakeholders* document

**Environment conditions**

This document is employed to describe the environment conditions in which the organization will be placed. It is an structured text document. Its template is shown in table 7 and table 8 gives an example of this work product. This document analyzes five conditions: the change rate, the complexity, the uncertainty, the receptivity and the diversity of an environment.

| **Environment conditions** |
|---|
| **Change rate**: Are the stakeholders constant through time? Are their requirements constant? Are they modified in a cyclical and a predictable way? Is it possible to estimate the consumption of a product? Is the demand of a product or a service constant through time? If the answer is affirmative, the environment is stable. If not, it is an unstable or dynamic environment. |
| **Complexity**: Is there a lot of different elements? Are there a lot of clients? Are there a lot of types of products and services to offer? Are there a lot of types of providers? Are providers not related between them? If any of the answers is affirmative, the environment is complex. If not, it is a simple environment. |
| **Uncertainty**: If the environment is dynamic and complex, uncertainty is high. If the environment is stable and simple, uncertainty is low. |
| **Receptivity**: Are the inputs and resources available? Are they obtained in an easy and secure way? If the answer is affirmative, the environment is munificent. If not, it is an hostile environment. |
| **Diversity**: Are different groups of clients served? Is it provided a set of different products or services, with no relationship between them? If any of the answers is affirmative, the environment is diverse. If not, it is a uniform environment. |

Table 7: *Environment Conditions* document

| *Environment conditions* | | |
|---|---|---|
| **Condition** | **Value** | **Justification** |
| Change rate | Stable | Stakeholders and activities remain unchanged trough time. |
| Complexity | Complex | There are a lot of clients, products and services. |
| Uncertainty | Medium | The environment of the organization is stable, but complex. |
| Receptivity | Munificent | The inputs and resources are obtained in an easy and controlled way from previously known providers. |
| Diversity | Diverse | There are a lot of types of clients, products and services. |

Table 8: Example of an *Environment Conditions* document

**Functional Dimension Model**

This work product is a GORMAS diagram. GORMAS uses an UML-like graphical notation called GOPPR[8] (used to define diagrams on INGENIAS and ANEMONA methodologies), but adding some entities proposed by GORMAS such as services and norms. A caption to understand the elements of the diagram is shown in figure 13.

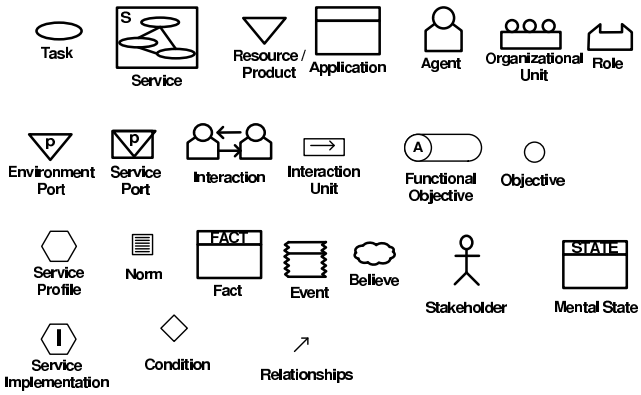



Figure 13: Entities from the GORMAS graphical notation

As stated before, the *Functional Dimension Model* details the specific functionality of the system, based on services, tasks and goals, as well as system interactions. In this phase of the methodology, the Organizational Unit representing the system, along with the stakeholders, the global goals, the products and the services of the system are depicted on this diagram. Figure 14 shows an example of a *Functional Dimension Model* diagram. An Organizational Unit is defined (UPV), containing two services (Budget management and PhD management) and two resources (Databases and Bills). The Organization pursues two objectives ('Efficient management of the financial resources' and 'Increase scientific production') and three groups of stakeholders (Students, Governing organs, Teachers) are consuming and offering services and resources.

## 2.2 Service analysis

In this phase (Figure 16), the services offered by the organization to its clients are specified, as well as how these services behave (inputs/outpus description and internal tasks) and which are the relationships (interdependencies) between these services. Furthermore, goals associated with services are detailed.

The *Service Analysis* phase involves (Figure 15) two different process roles, five work products (three GORMAS models diagrams and two text documents) and one guidance related to the technology of the organization. This step is composed by five tasks (Organization Technology Analysis, Organizational Unit Technology Analysis, Work Flow and Technological Interdependence Analysis, Define Organization Functionality and Analysis of

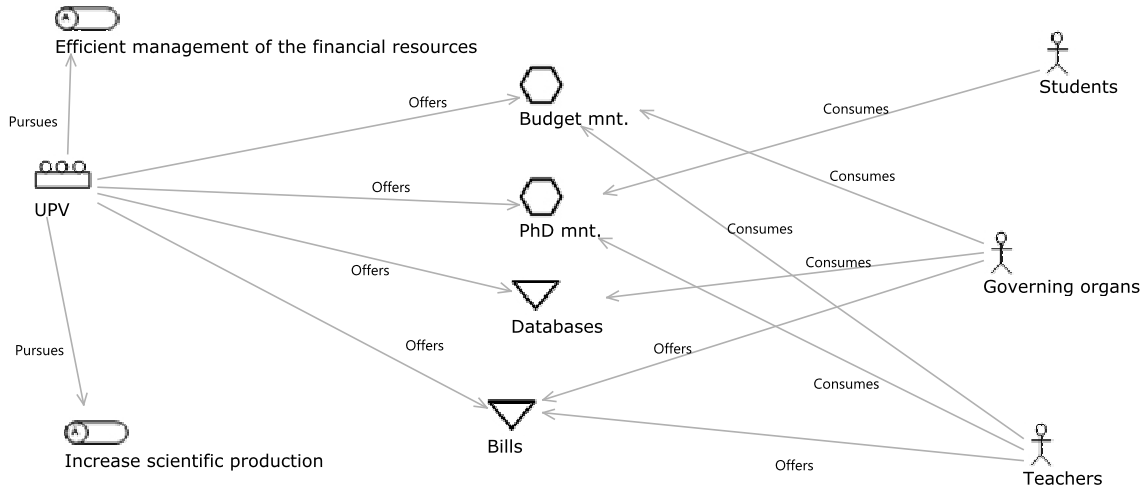



Figure 14: Example of a *Functional Dimension Model* diagram

the goals of the organization). The flow of tasks of *Service Analysis* phase is shown in figure 16

Taking the Organization Theory as a basis, three existing types of technology are considered: (i) the *Organization Technology*, which refers to the whole organization and determines in which measure does any client have influence in the process of production and the final aspect of the product or in which measure the services are related between them and with regards to the clients; (ii) the *Organizational Unit Technology*, which contemplates the diversity and complexity of the different organizational tasks, identifying the existing flows of work; and (iii) *Work Flow and Technological Interdependence*, that defines the interdependent relations originated as a result of the flow of work between the units of the organization.

As a result of this phase, the diagrams of the *Structural* and *Environment Dimension Models* are generated and the *Functional Dimension Model* diagram is updated. Specifically, in the *Functional Dimension Model*, both resources and applications of the system are identified. In the *Structural Dimension Model* the entities representing the clients or providers of the system that are required to participate inside are established. Moreover, the services required and offered by the system are identified, as well as the roles that provide or make use of these services. For each service, the *Functional Dimension Model* diagram must be updated, detailing its profile (inputs, outputs, preconditions and postconditions) and its tasks. Finally, in the *Functional Dimension Model* the mission is split into functional goals, which are related to the system entities and its services. The functional goals represent the specific actions of the organizational units and their expected results.

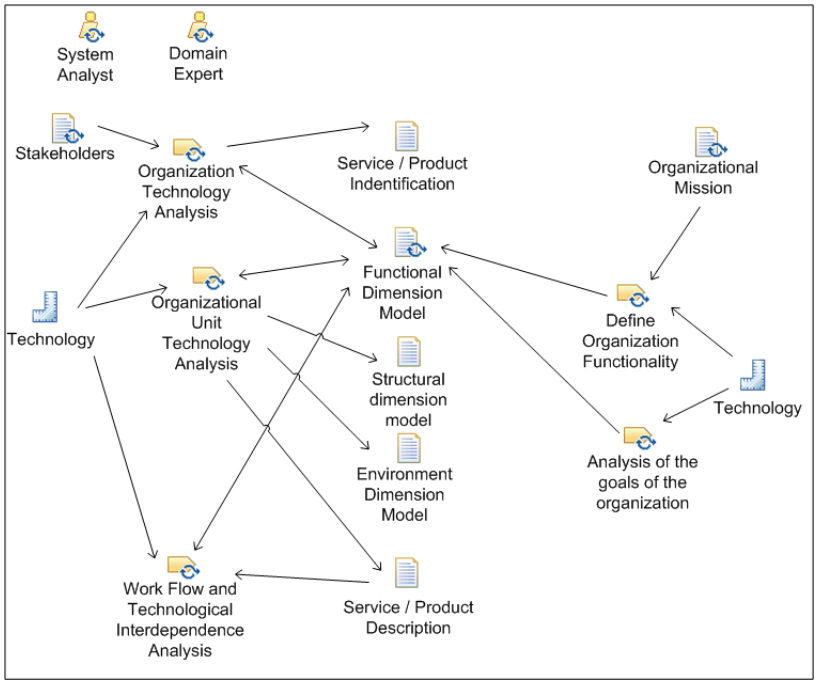



Figure 15: Resources and products used in *Service Analysis* phase

### 2.2.1 Process roles

There are two roles involved in this phase: the role of the System Analyst and the role of the Domain Expert. Both of them are described in the following subsections.

**System Analyst**

He is responsible of:

- Analyzing the technology of the system, along with the goals of the organization.
- Identifying and describing the products and services of the organization.
- Splitting the mission goals of the organization into functional goals.
- Modeling the structural and the environment dimension model diagrams.
- Updating the functional dimension model.

**Domain Expert**

He is in charge of supporting the System Analyst on the analysis of the technology, the goals, the services and the products of the system.

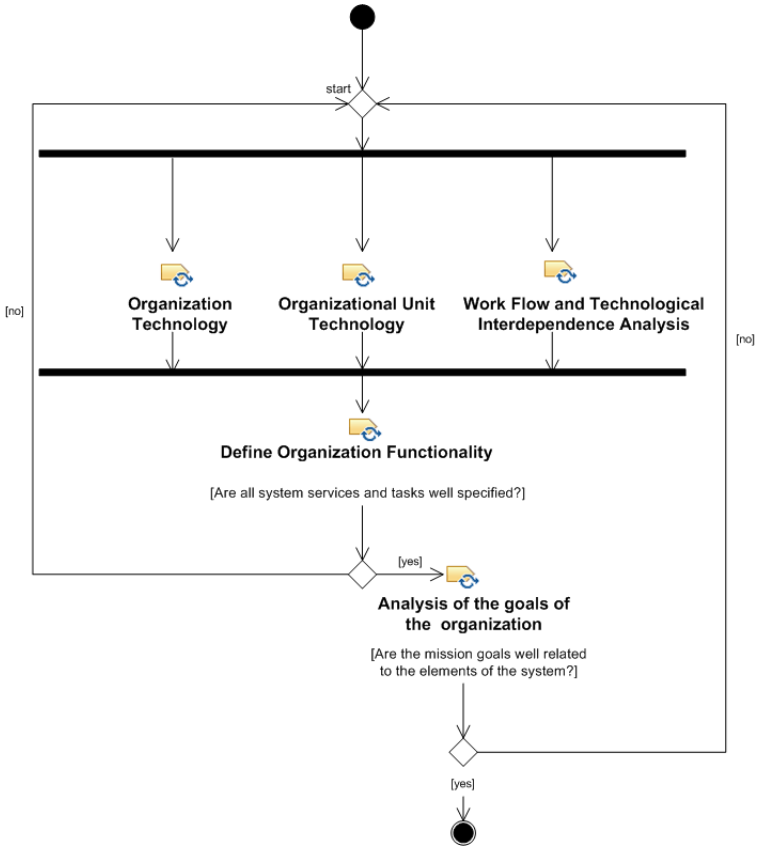



Figure 16: Activity diagram of *Service Analysis* phase

### 2.2.2 Activity details

In multiagent system domain, the technology concept refers to the set of resources, applications and knowledge required by agents, as well as the set of processes and tasks that are necessary to carry out with the services offered by the organization. This step is aimed to identify the kind of technology that the organization will use. Moreover, the designer will describe in detail the products generated and the services provided by the organization.

Besides of the description of services and products, there is another main activity in this step of the methodology. The mission of the system must be derived into other goals or objectives: functional objectives and operative objectives. Functional objectives represent the results that organizational units are expected to achieve. The operative objectives are the specific and measurable results which are expected to be achieved by the members of a unit. Therefore, this activity aims to split the goals of the system into functional objectives and to set their relationships with organizational units, A-Agents and services of the organization. Tasks of this activity are detailed in table 9.

More specifically, at the *Service Analysis* phase it is specified:

- The type of products and services that the system offers to or consumes from its stakeholders.
- The tasks related to the production of the products and services, defining the steps necessary for obtaining them, their relationships and interdependences between the different services and tasks.
- The goals related with the achievement of these products or services.
- The resources and applications needed for offering the system functionality.
- The roles related with the stakeholders, on the basis of the type of services or tasks that they provide or consume.

| **Task** | **Task Description** | **Roles Involved** |
|---|---|---|
| Organization Technology Analysis | Determine the way in which products and services are produced. It could be "organization directed", "client directed" or "standard production". | System analyst and domain expert |
| Organizational Unit Technology Analysis | Existing workflows are determined and services and products are analyzed using the templates provided by the methodology. | System analyst and domain expert |
| Work Flow and Technological Interdependence Analysis | Identify what is the relationship between organizational units in order to reach the organizational goals. There are three kinds of interdependence: independent (coordination between units must be minimum), sequential (tasks are linked or follow a required workflow) and reciprocal (units depend of each other). | System analyst and domain expert |
| Define organization functionality | It must be checked whether services and their related tasks are well specified. Additionally, for every A-Agent or Organizational Unit identified in the functional dimension model, it is assigned a split of the mission goals, by means of the products it generates, the clients that group or the services it attends. | System analyst |
| Analysis of the goals of the organization | For every identified Service, a link is designed between one or some functional objectives and its Organizational Unit. | System analyst |

Table 9: Tasks of the *Service Analysis* step

### 2.2.3 Work products

The following section describes the products generated by the *Service Analysis* phase. Two models are defined: the *Structural* and *Environment Dimension Model* diagrams. Moreover, the *Functional Dimension Model* is updated. Two kinds of documents are defined: the *Service / Product Identification*, which is a structured text document that allows defining the type of production of the organization and the kind of technology that the system will use (see table 10); and the *Product / Service Description*, a document describing the features of the products and services of the organization. Figure 17 describes their relation with the elements of the GORMAS metamodel.

| **Name** | **Description** | **Work Product Kind** |
|---|---|---|
| Functional Dimension Model | A diagram using the GORMAS graphical notation (based on GOPPR notation) that details the specific functionality of the system, based on services, tasks and goals. | Behavioral |
| Structural Dimension Model | A diagram that uses the GORMAS notation that describes the components of the system and their relationships. It allows defining the static components of the organization. | Structural |
| Environment Dimension Model | A diagram that uses the GORMAS notation that describes the environment elements (resources and applications), along with the agents' behavior. It also allows defining the service ports. | Behavioral |
| Product / Service Identification | A document that identifies the kind of production the system will have and the technology that products will use. | Structured text |
| Product / Service Description | A document that describes the features of the products and services. | Structured text |

Table 10: Products for *Service Analysis* phase

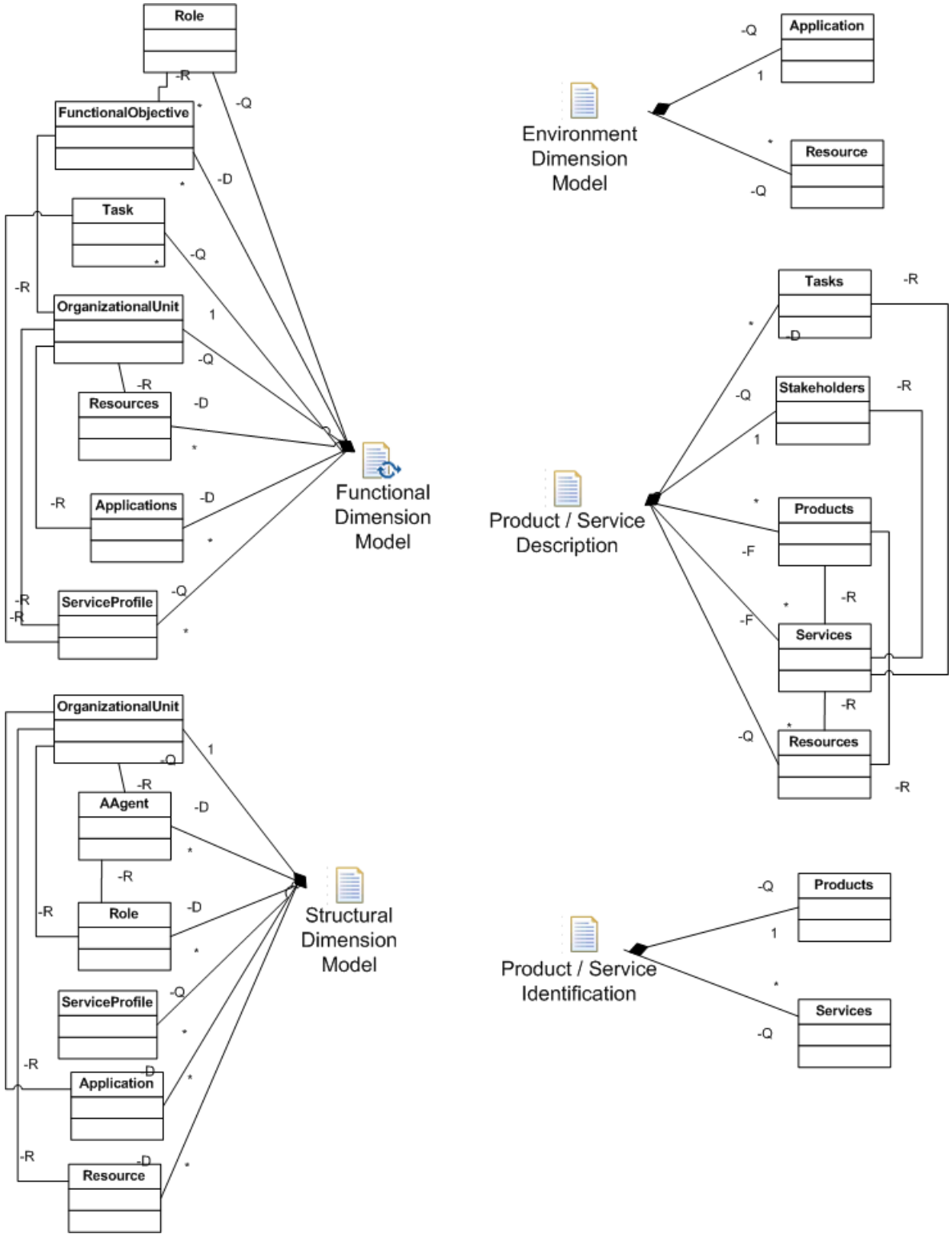



Figure 17: *Service Analysis* phase. Relations between work products and metamodel elements. **Caption**: D: element introduced for first time; F: element refined; Q: element already defined; R: element related with another element.

**Product / Service Identification**

The *Product / Service Identification* document is a couple of structured text documents that describe the technology that the system will use. On the one hand, the technology used to obtain the products is identified. It could be *organization directed*, *client directed* or *standard production*, depending on who is requesting the products. On the other hand, the technology followed by the service organization is described. Services could have *inter-dependence, dependence* or *variability* among them. The template of this document can be found in tables 11 and 12; an example is shown in table 13.

| **Product identification** |
|---|
| Are the obtained products used as resources to make new products? Is the production continuous without a clear start and a clear end? Are modules from other processes used? Are the modules assembled in order to obtain products under request?<br>In case of affirmative answer, then it is an **organization directed** production.<br>• *Tip*: Basic products that are components to make more complex products must be identified. They must be represented as resources. |
| Do products fit the client's needs? Is every product made for a concrete client, under his request? Is a wide range of products managed? Are a big variety of the clients' requirements taken in account? Are frequent changes on products features anticipated?<br>In case of affirmative answer, then it is a **client directed** production.<br>• *Tip*: Every client must be represented by an A-Agent. It must be included as a member of the Organizational Unit that represents the system. |
| Could a product be consumed by different kinds of clients? Could a product be consumed by different clients that belong to the same type? Is their production independent from the requirements of their final consumers?<br>In case of affirmative answer, then it is an **standard** production.<br>• *Tip*: Applications for interacting with the clients must be created. The Organizational Unit that represents the system must contain at least one application per client. All of these applications will allow extracting information related to specific clients. |

Table 11: *Product Identification* document

| **Service identification** |
|---|
| Is the functionality of every service independent? Is the order that the services are executed making no difference? Are some types of clients connected by means of services?<br>In case of affirmative answer, then it is **service independence**. |
| Is a determined order to execute the services required? Are the inputs of a service depending from the outputs of another service? Does a client need to use a previously established service to request another service?<br>In case of affirmative answer, then it is **service dependence**. |
| Is the order of the services to offer variable? Does it depend on the decisions and requirements of the clients? Are the needs of the clients unpredictable?<br>In case of affirmative answer, then it is **service variability**. |

Table 12: *Service Identification* document

| *Product / service identification* |
| --- |
| **Product technology:** *client directed production.* Products are made by means of the requirements of the clients of the organization. |
| **Service technology:** *service dependence.* Services are executed in a specific order. Outputs of a service could be inputs for another service. |

Table 13: Example of a *Product / Service Identification* document

**Product / Service Description**

The *Product / Service Description* document is a couple of structured text documents that describe the products and services that the organization will produce. Services are described by means of their functionality, the roles involved on them and their profile. For product description, it is necessary to specify their granularity, their margins, their parameters and the resources that they will use. The template of this document can be found in tables 14 and 15; an example is shown in table 16. The syntax employed in the *Service Description* document for describing tasks is included in table 17.

| *Organizational Unit Technology* | |
|---|---|
| **Name** | *Name of the product* |
| **Description** | A short description about the product. |
| **Parameters** | A set of the product features that will be considered. |
| **Lower margin** | Minimum value used for the service. |
| **Upper margin** | Maximum value used for the service. |
| **Granularity** | Grade of variability that the values for every parameter could have. |
| **Resources** | A set of environment entities that the product will use. |
| *Tip* | If a "client directed" production is used, then variables defined by the clients must be set. |

Table 14: *Product Description* document

| *Organizational Unit Technology* | |
|---|---|
| **Name** | *Name of the service* |
| **Description** | A short description about the service functionality. |
| **Conditions** | |
| *Context* | An specification of the environment where the service is executed. |
| *Exceptions* | A set of conditions that prevent the correct execution of the service. |
| **Consumer** | An actor that requests the service. |
| Objective | What the consumer is looking for by using the service? |
| Price | The value that the consumer should pay while using the service. |
| Benefit | A description of the benefit obtained by the consumer. |
| **Producer** | An actor entrusted to provide and execute a service. |
| Objective | What the producer is looking for by providing the service? |
| Cost | A description of the cost to pay while using the service. |
| Benefit | A description about the benefit obtained while providing the service. |
| **Service Profile** | |
| *Inputs* | Information that must be supplied to the service. |
| *Preconditions* | A set of the input conditions and some environment values in order to obtain a correct execution of the service. |
| *Outputs* | Information returned by the service. |
| *Postconditions* | Final states of the parameters of the environment, by means of the different kinds of outputs. |
| **Functionality** | |
| *Tasks* | A set of tasks covered by the service. It must be pointed out whether any of these tasks is another service provided by the organization. |
| *Resources* | A set of environment entities that the service will use. |
| *Provider* | An actor that provides the given resources. |
| *Products* | Tangible results obtained by executing the service. |

Table 15: *Service Description* document

| *Organizational Unit Technology* | |
|---|---|
| **Service** | *Payments and incomes validation* |
| **Description** | A service to validate the payments and incomes the organization must afford. |
| **Conditions** | |
| *Context* | Payment proposals are evaluated every month. |
| *Exceptions* | There is not the correct day to start the service. |
| **Consumer** | Governing organism |
| Objective | To make financial operations concerning payments and incomes. |
| Price | To send the balance and the amount in the correct way. |
| Benefit | Received income. |
| **Producer** | Technical staff |
| Objective | To control that the process is correctly carried out. |
| Cost | |
| Benefit | |
| **Service Profile** | |
| *Inputs* | – Monthly balance= ⟨IDMonBal, Amount⟩ |
| | – Salary =⟨IDSalary, amount⟩ |
| *Preconditions* | ¬ ∃ a ϵ Staff Service DB \| a.IDProposal=IDProposal |
| *Outputs* | Report: balance+refunds=⟨IDReport,balance,refunds⟩ |
| *Postconditions* | ∃ a ϵ Staff Service DB \| a.IDProposal=IDProposal ∀ Proposal |
| **Functionality** | |
| *Tasks* | MonBalChe.FinDocProd$^n$.CalTot.FinDocProd |
| | – FinDocProd: Finantial documents production |
| | – MonBalChe: Monthly balance checking |
| | – CalTot: Calculate totals |
| *Resources* | Proposals and deposits lists=⟨IDList, proposals*,deposits*⟩ |
| | Salary=⟨IDNomina⟩ |
| | Staff Service DB |
| *Provider* | Applications |
| | Educational center |
| | Technological transfer center |
| | Staff service |
| *Products* | Balance report=⟨IDReport⟩ |
| | Financial Documents=⟨IDDocument,type⟩ |
| | type={DCRK, CK, MC/, MC, INP, ADOK, RC, OK, P, T} |

Table 16: *Service Description* document example

| **Task flow syntax** | |
|---|---|
| **Operator** | **Meaning** |
| $x.y$ | $x$ followed by $y$ |
| $x \mid y$ | $x$ or $y$ occurs |
| $x^n$ | $x$ occurs $n$ times |
| $x^+$ | $x$ occurs one or more times |
| $x^*$ | $x$ occurs zero or more times |
| $[x]$ | $x$ is optional |
| $x \mid\mid y$ | $x$ and $y$ are simultaneous |

Table 17: Syntax employed to describe the task flow in the *Service Description* document

**Functional Dimension Model**

The *Functional Dimension Model* diagram was defined on *Mission Analysis* step (see section 2.1.3). In this step, roles are related with the service they offer or require, services are split into tasks (defining the resources and applications that they offer and consume) and the mission of the system is split into functional objectives. As an example, figure 18 shows the tasks in which the service PhD Management is split on. Figure 19 depicts the mission of the system split on functional objectives. This split forms the Goal Tree of the organization.


Figure 18: Service *PhD Management* split in tasks

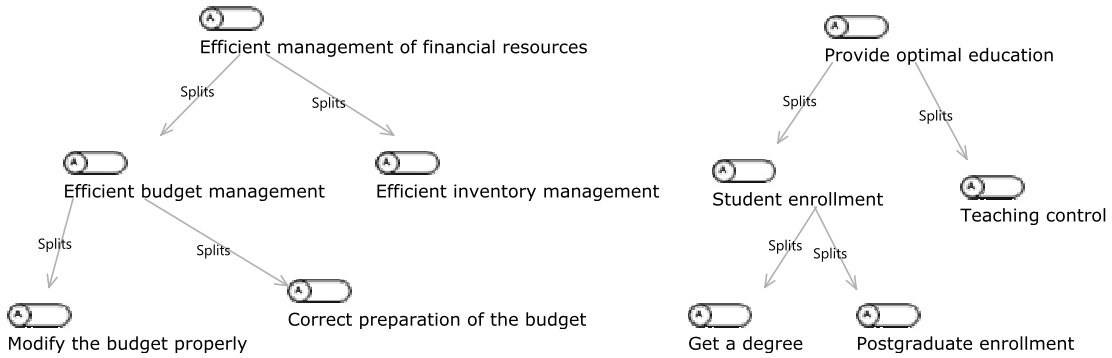



Figure 19: Relationship between mission and Functional Objectives of the organization

**Structural Dimension Model**

As stated before, the *Structural Dimension Model* describes the components of the system and their relationships, allowing the definition of the static components of the organization. In this phase of the methodology, the Organizational Unit representing the whole system is depicted, as well as all those entities that have been identified in the previous steps. Figure 20 shows an example of an *Structural Dimension Model* diagram in which a "client directed" technology has been considered. In this figure, three agents (Student, Teacher and Governing Organ) and three roles with the same names as agents are defined along with the OU (UPV).

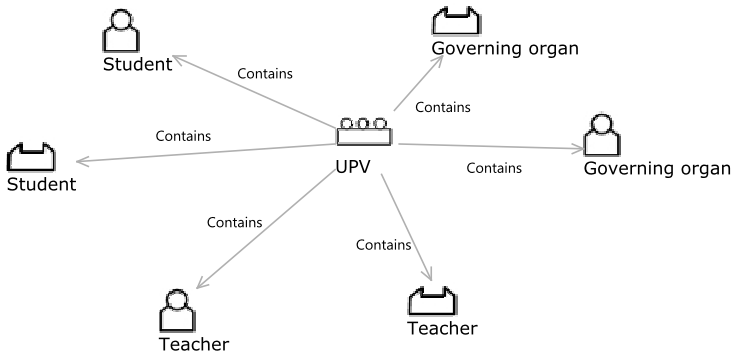



Figure 20: Example of a *Structural Dimension Model* diagram

**Environment Dimension Model**

As previously explained, the *Environment Dimension Model* describes the elements of the environment of the system (resources and applications). In this phase of the methodology the resources and applications used by the organization are depicted. Figure 21 shows an example of an *Environment Dimension Model* diagram, showing an OU (UPV) which contains two resources (Database and Bills) and an application (Calculate Totals).

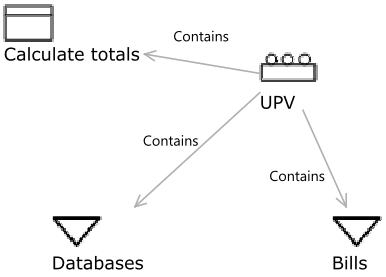



Figure 21: Example of a *Environment Dimension Model* diagram

## 2.3 Organizational Design

In this development step (Figure 22), the structure most adapted for the Virtual Organization is selected. This structure will determine the relationships and pre-established restrictions that exist between the elements of the system, based on specific dimensions of the organization, which impose certain requirements on the types of work, on the structure of the system and on the interdependence between tasks.

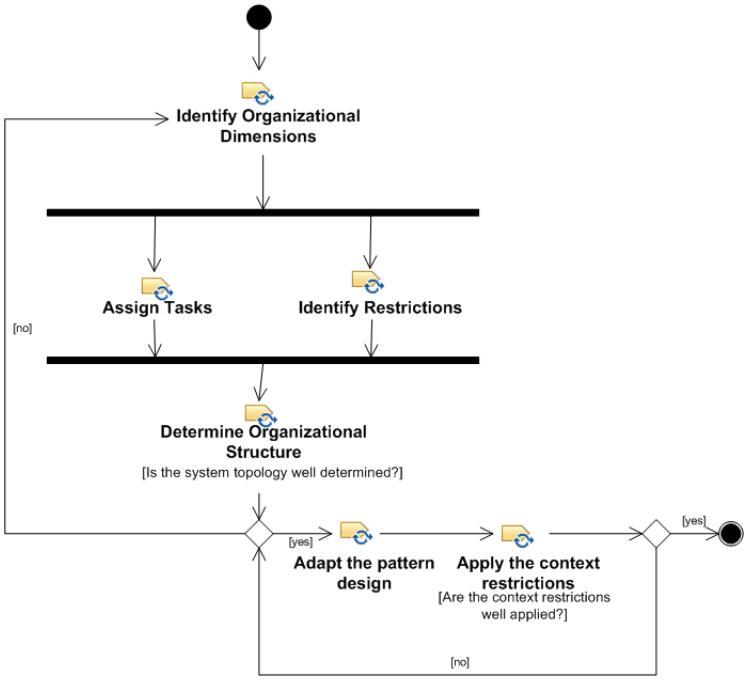



Figure 22: Activity diagram of *Organizational Design* phase

For the structure selection, a decision-tree (Figure 23) has been elaborated, that enables the system designer to identify which is the structure that better adjusts to the conditions imposed by the organizational dimensions. This decision-tree can be applied to the system as a whole or to each of its OUs, so then enabling structure combinations. At the end of this step of the methodology, context restrictions identified on Mission Analysis phase will be taken into account in order to modify the structure of the organization if necessary.

Additionally (see figure 24), as a result of this phase, *Functional and Structural Dimension models* are updated. *Structural dimension model* is updated by adding new Organizational Units, Roles, Resources and Norms. *Functional Dimension Model* is updated in order to show new relationships between Roles and Organizational Rules and to describe the functionality of the services.

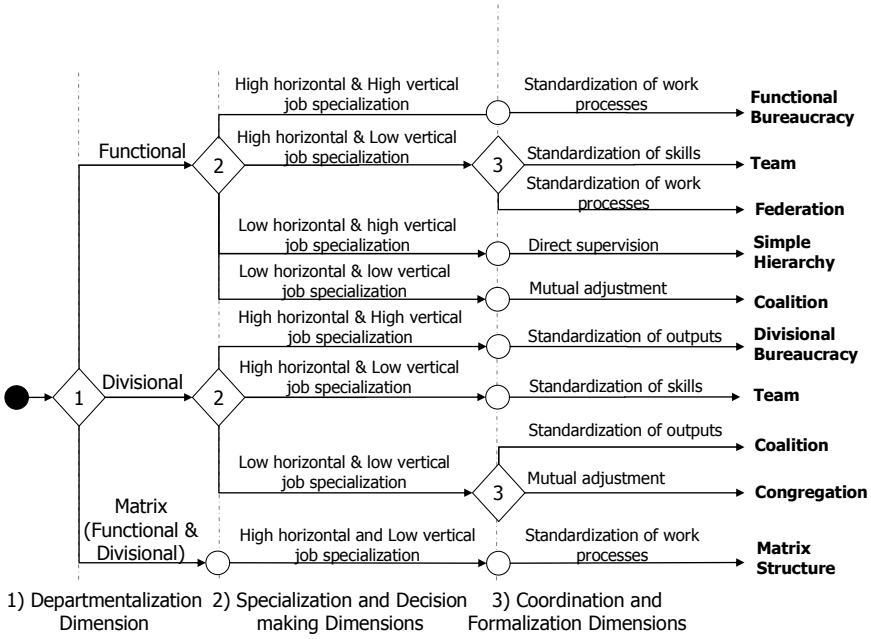



Figure 23: Organizational Structure decision-tree

### 2.3.1 Process roles

There are two roles involved in the Organizational Design phase: the System Analyst and the Domain Expert. Both of them are described in the following subsections.

**System Analyst**

He is responsible for:

- Defining the organizational dimensions of the system.
- Selecting the most suitable structure for the organization.
- Adapting the selected structure to the organization.

**Domain Expert**

He is responsible for supporting the system analyst during the Organizational Design phase, by giving him all the information that he could need about the organizational dimensions of the system.

### 2.3.2 Activity details

As stated before, the first task to be accomplished in this phase is to identify the specific dimensions of the organization. These organizational dimensions are:

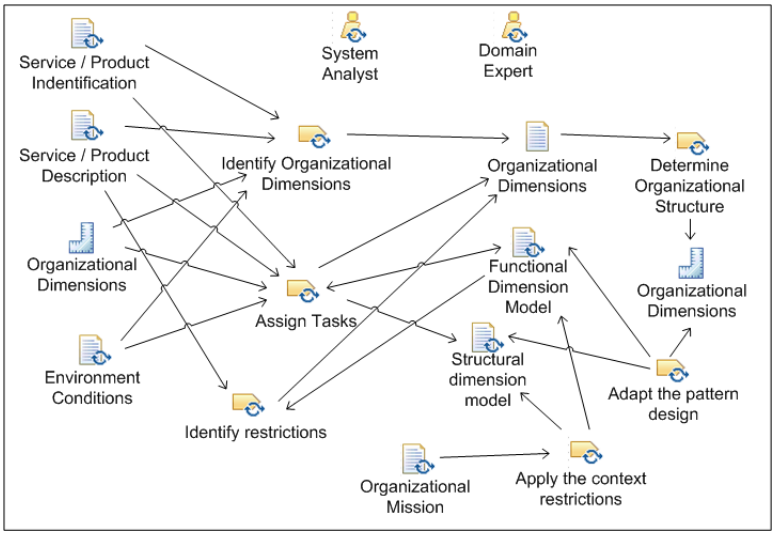



Figure 24: Products of *Organizational Design* phase

- *Departmentalization*, which details the motivation of work group formation, i.e. functional (on the basis of knowledge, skills or processes) or divisional (on the basis of the characteristics of the market, clients, products or services).
- *Specialization*, which indicates the degree of task division, based on the quantity and diversity of tasks (horizontal job specialization) and the control exercised on them (vertical job specialization).
- *Decision making*, that determines the degree of centralization of the organization, i.e. the degree of concentration of authority and capture of decisions.
- *Formalization*, which specifies the degree in which tasks and positions are standardized, by means of norms and rules of behavior.
- *Coordination Mechanism*, which indicates how individuals can coordinate their tasks, minimizing their interactions and maximizing their efficiency, using mutual adjustment, direct supervision or standardization.

These organizational dimensions are employed for determining the most suitable structure for the system specifications. Thus, it is carried out:

- The analysis of the organizational dimensions, which allows specifying the functional granularity of the service, by means of grouping tasks and services together (defining more complex services) and assigning them to specific roles; and identifying general restrictions on the coordination and cooperation behavior of the members of the organization.

- The selection of the structure of organization most interesting for the system.

- The adaption of the selected structure to the problem under study, using specific design patterns.

Along with the decision-tree, a set of design patterns of different structures has been defined which include simple hierarchy, team, flat structure, bureaucracy, matrix, federation, coalition and congregation structures. These patterns describe their intrinsic structural roles, their social relationships, as well as their typical functionality. According to these design patterns, the diagrams of the organizational and activity models are updated, so then these intrinsic roles, relationships and functionality, related to the design pattern are integrated inside the current problem.

At the end of this phase, the context restrictions identified on *Mission Analysis* phase (*Organizational Mission* document) are taken into account to provide a final design of the organization. Therefore, the structure obtained after integrating the pattern design must be modified to satisfy the context restrictions. This is an iterative process. First of all, textual restrictions are needed to be transformed into GORMAS entities. These entities can be Organizational Rules, Roles, Agents, Services, Norms, etc. For example, if one of the restrictions specifies 'There must be a manager', a role called 'Manager' must exist into the organization. Next, it is necessary to integrate these restrictions into the selected structure, that must be adapted until an optimal structure for the organization is achieved.

All tasks of this activity are detailed in table 18.

| **Task** | **Task Description** | **Roles Involved** |
|---|---|---|
| Identify Organizational Dimensions | Define the organizational dimensions of the system by assigning tasks and identifying restrictions. | System analyst and Domain Expert |
| Assign tasks | Tasks are grouped into Organizational Units and assigned to members of the organization. | System analyst and domain expert |
| Identify restrictions | Existing restrictions about the organization members behavior are identified. Additionally, mechanisms to help coordination and cooperation between members are described | System analyst and domain expert. |
| Determine Organizational Structure | Using the organizational dimensions previously identified, along with the decision tree (see figure 23), the best structure for the organization is identified. This activity can be applied not only to the whole organization, but also to some OUs. | System analyst |
| Adapt the pattern design | The identified structure is adapted by modifying Functional and Structural Dimension Models | System analyst |
| Apply the context restrictions | Context restrictions identified on Mission Analysis phase are applied to the structure of the organization. | System analyst |

Table 18: *Organizational Design* activity

### 2.3.3 Work products

The following section describes the products generated by the organizational design phase. Two models are updated: the *Structural* and *Functional Dimension Model* diagrams. Moreover, the *Organizational Dimensions* structured text document is defined. It describes the specific dimensions of the organization that will be used to select the structure that best fits the system. Work products are described in table 19. Figure 25 describes their relation with the elements of the GORMAS metamodel.

| **Name** | **Description** | **Work Product Kind** |
|---|---|---|
| Functional Dimension Model | A diagram using the GORMAS graphical notation (based on GOPPR notation) that details the specific functionality of the system, based on services, tasks and goals. | Behavioral |
| Structural Dimension Model | A diagram that uses the GORMAS notation that describes the components of the system and their relationships. It allows defining the static components of the organization. | Structural |
| Organizational Dimensions | A document that describes the specific dimensions of the organization: *departmentalization*, *specialization*, *decision making*, *formalization* and *coordination mechanism*. | Structured text |

Table 19: Products for *Organizational Design* phase

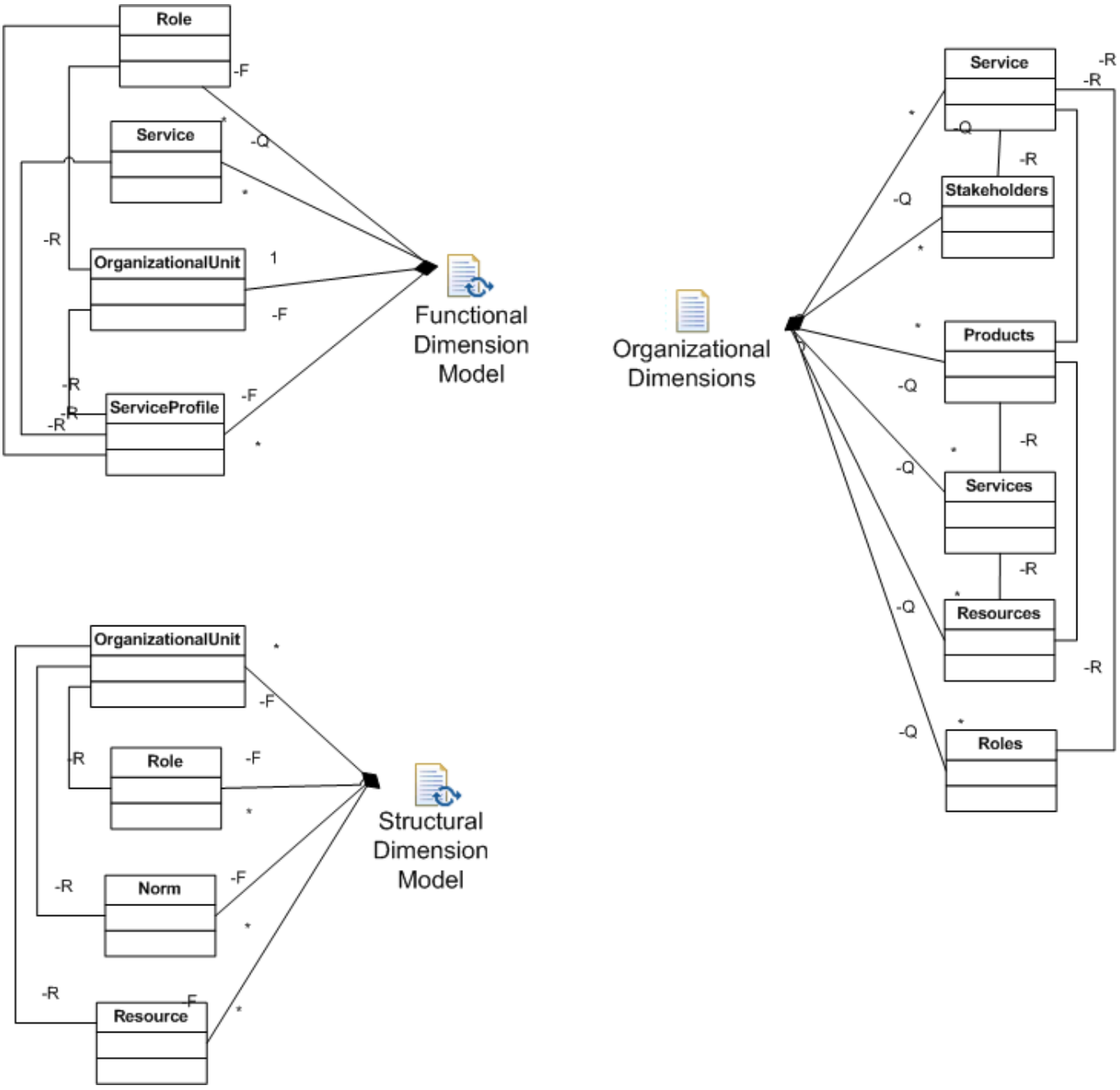



Figure 25: Organizational Design phase. Relations between work products and metamodel elements. **Caption**: F: element refined; Q: element already defined; R: element related with another element.

### Organizational Dimensions

The *Organizational Dimensions* document is a structured text document that describes the dimensions that impose some requirements to the organization. These dimensions are *departmentalization*, *specialization*, *decision making*, *formalization* and *coordination mechanism*. The template of this document can be found in table 21 and an example in table 20.

| **Organizational Dimensions** |
|---|
| **Departmentalization**: Functional. Services are grouped by means of their functionality. |
| **Specialization and Decision making**: High horizontal and low vertical job specialization.<br>Roles use and provide a reduced set of types of services. These roles are specialized, but they do not have enough control over the services. |
| **Coordination and Formalization**: Standardization of work processes.<br>Order of tasks is very important and each task must be performed by a specific agent. |

Table 20: *Organizational Dimensions* document example

| **Organizational Dimensions** |
|---|
| **Departmentalization** |
| Is it necessary to act on the same type of resources? Does the offered functionality follow similar behavior patterns? Do the services require the same kind of inputs? Do they offer similar outputs?<br>In case of affirmative answers, it is a **functional departmentalization**.<br>• *Tip*: Similar functionalities must be grouped and an Organizational Unit must be created for every group. |
| Do activities must be particularized by means of the clients or the offered products? Is the functionality different according to the geographical location?<br>In case of affirmative answers, it is a **divisional departmentalization**.<br>• *Tip*: Functionalities aimed to the same client must be grouped and an Organizational Unit must be created for every group. |
| Are similar functionalities for different kinds of clients or products available? Are these functionalities related between them, following a specific order?<br>In case of affirmative answers, it is a **functional and divisional departmentalization**. |
| **Specialization and Decision Making** |
| Have the roles got specialized tasks assigned? Do the roles make few types of tasks? Are the roles lacking of control over their own work?<br>In case of affirmative answers, it is a **high horizontal and high vertical job specialization**.<br>• *Tip*: Agents' tasks control will be assumed by supervisor agents, that are in charge of their Organizational Units. |
| Do the roles have assigned specialized tasks but exerting control over them? Can agents select the mechanism to carry out these tasks?<br>In case of affirmative answers, it is a **high horizontal and low vertical job specialization**. |
| Are the roles in charge of different tasks a bit related among them? Have these roles a low interdependence? Do the roles not offer control over their activities?<br>In case of affirmative answers, it is a **low horizontal and high vertical job specialization**. |
| Are the roles in charge of some different tasks, assuming their control?<br>In case of affirmative answers, it is a **low horizontal and low vertical job specialization**. |
| Is the environment simple? Is it necessary to process few quantities of information?<br>In case of affirmative answers, centralization is recommended. |
| Is the environment complex? Is it necessary to process big amounts of information? In case of affirmative answers, centralization is not recommended. |
| **Coordination and Formalization** |
| Is the environment dynamic? Is the way to execute tasks flexible? Can tasks be carried out by different roles? Do roles present low vertical job specialization? **Mutual adjustment** (negotiation processes) must be applied. |
| Do the roles present high vertical job specialization? Are the control and management centralized in some points? **Direct supervision** must be applied. |
| Is the order of execution of tasks very important? Is it more indifferent to the resolution method by which the system has been obtained? Is it indifferent of who is in charge of providing? **Standardization of skills** is applied. |
| Are the knowledge and skills that are available on certain tasks indispensable? Is there a predefined and globally assumed behavior for a particular skill? **Standardization of skills** must be applied. |

Table 21: *Organizational Design* document

**Functional Dimension Model**

The *Functional Dimension Model* diagram was defined on *Mission Analysis* step (see section 2.1.3) and updated on *Service Analysis* step (see section 2.2.3). In the *Organizational Design* step, *Functional Dimension Model* is updated by adding relationships between roles and the services they provide/consume, along with the relationships between Organizational Units and roles, and the services they provide. Additionally, services must be split into services. An example of an updated *Functional Dimension Model* is shown in figure 26. This figure shows two split OUs (Teaching and Economy) and two split services (Request diploma and Request budget), along with the Contains relationships between OUs and Roles and it shows which roles are consuming or offering the services.

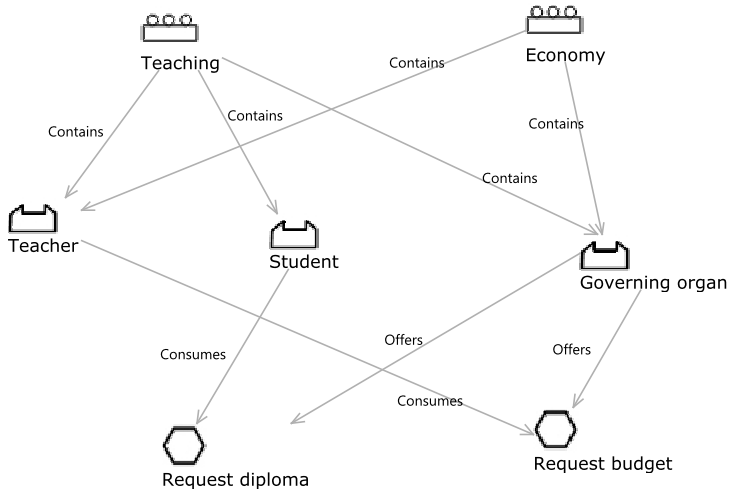



Figure 26: Example of a *Functional Dimension Model* diagram updated on *Organizational Design* phase

**Structural Dimension Model**

The *Structural Dimension Model* diagram was defined on *Service Analysis* step (see section 2.2.3). In the *Organizational Design* step, *Structural Dimension Model* is updated by adding new Organizational Units, roles and their relationships, and by relating OUs with the roles they contain. It is necessary to specify the inheritance of roles. Additionally, entities defined by the design pattern must be added to the diagram and the structure that best fit the organization is adopted. An example of an updated *Structural Dimension Model* is shown in figure 27. This figure depicts two OUs (Teaching and Economy) contained into the OU that represents the system (UPV) and five roles (Student, PhD Student, Professor, Teacher and Governing organ), also representing the inheritance relationships between roles.

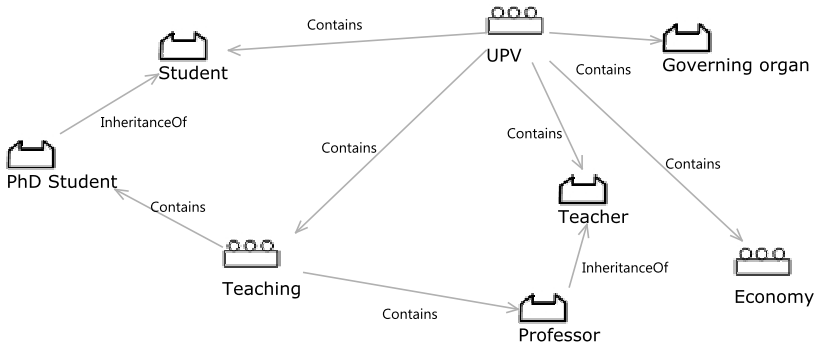



Figure 27: Example of a *Structural Dimension Model* diagram updated on *Organizational Design* phase

## 2.4 Organization Dynamics Design

In this activity (Figure 28), the detailed design of the system is carried out, which implies the following activities: the design of the information-decision processes; the design of the system dynamics as an open system; the design of the measuring, evaluation and control methods; the definition of a suitable reward system; and finally, the design of the system agents, describing them by means of diagrams of the agent model.

This step involves one process role and twelve work products (five model diagrams and seven text documents). This phase is composed of four activities (Design of Information-Decision Processes, Design of Open System Dynamics, Design of Control Policies and Design of the Reward System).

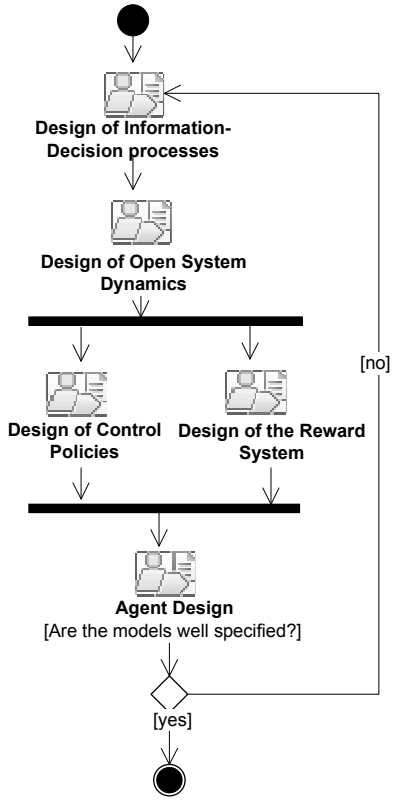



Figure 28: Activity Diagram of *Organization Dynamics Design* phase

### 2.4.1 Process role

There is one role involved in the Organizational Design phase: the System Analyst. He is responsible for:

- Detailing services and related workflows.

- Identifying the operative goals.
- Checking the procedures for obtaining information of the environment.
- Defining permissions for accessing resources or applications.
- Defining the specific interactions between agents and their collaboration diagrams.
- Defining the ontology of the domain.
- Determining the services that have to be advertised.
- Determining the policies for role enactment.
- Identifying the internal and external agents.
- Considering standardization of work process, outputs and skills.
- Analyzing the types of behavior needed to promote.
- Selecting the type of reward system to be used.
- Applying the selected reward system in the specific domain problem.

### 2.4.2 Activity details

The activities that compose this phase of the methodology are detailed as follows.

**Design of Information-Decision Processes**

The flows of information and adoption of decisions are described in order to determine how the information is processed and how agents work for obtaining the expected results (Figure 29). More concretely:

- The concrete functionality of services is specified, which implies:
  - Detailing services and related workflows for providing these services and splitting these workflows in concrete tasks.
  - Identifying the operative goals, that represent the specific and measurable results that the members of an OU are expected to achieve. Thus, functional goals are split into operative goals, which are lately assigned to tasks.
- The flow of information of the system is specified, which implies:
  - Checking the procedures for obtaining information of the environment, i.e. verifying whether there exists a contact point with the system for any stakeholder, using resources, applications or representative agents.

– Defining permissions for accessing resources or applications.
– Defining the specific interactions between agents on the basis of services and their collaboration diagrams, in which agent messages are defined.
– Defining the ontology of the domain, whose concepts will be used in the interactions, taking into account the service inputs and outputs.

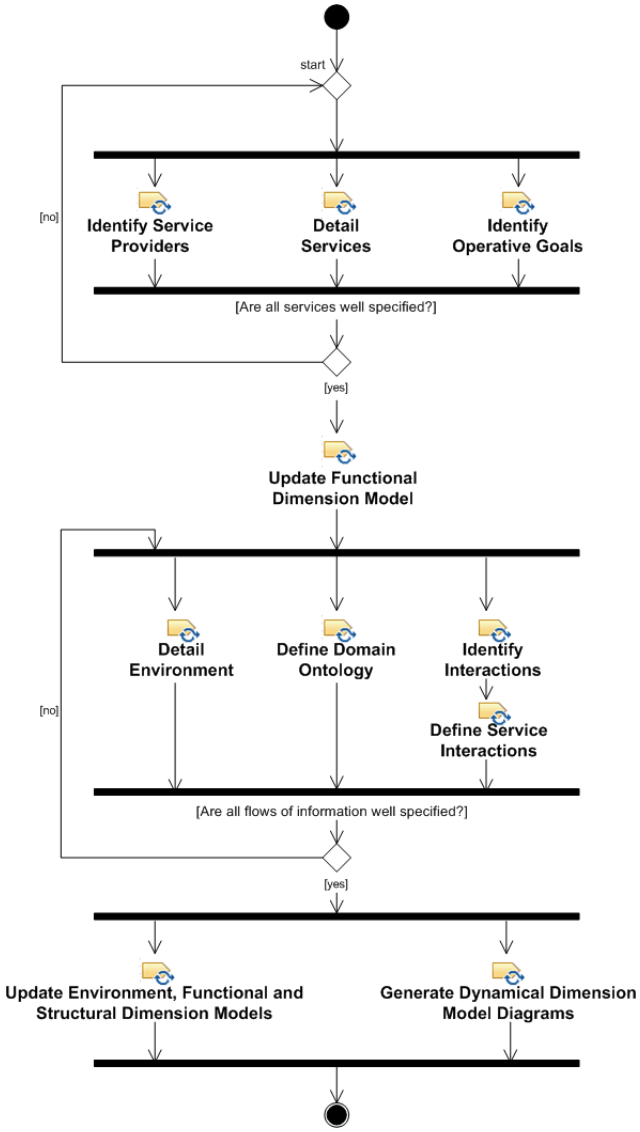



Figure 29: Activity Diagram of *Design of Information-Decision Processes* activity.

As a result (Figure 30), the diagrams of the *Dynamic Dimension Model* are defined. Moreover, *Environment*, *Structural* and *Functional Dimension Model* diagrams are updated. All tasks of this activity are detailed in table 22.

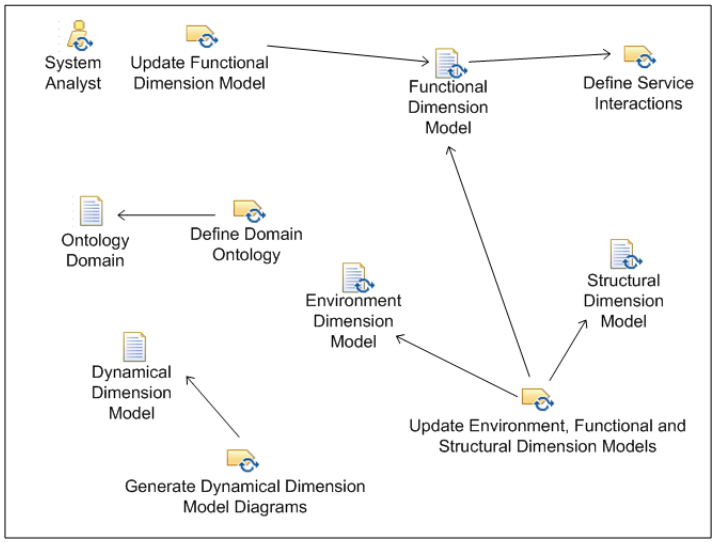



Figure 30: Products of the *Design of Information-Decision Processes* activity

**Design of Open System Dynamics**

In this activity (Figure 31), the functionality offered by the Virtual Organization as an open system is established, which includes the services that must be advertised as well as the policies of role enactment. In this sense, it is specified which is the functionality that must be implemented by the internal agents of the system and which functionality must be advertised in order to enable this functionality to be provided by external agents. Therefore, it is determined:

- The services that have to be advertised.
- The policies for role enactment, detailing the specific tasks for the *AcquireRole* and *LeaveRole* services of each OU.
- The identification of the internal and external agents.

All roles that need a control of their behaviors require a registration process inside the OU in which they take part, so they are associated with external agents, who must request the *AcquireRole* service to play this role. On the contrary, roles like managers or supervisors are assigned to internal agents, since their functionality needs of sufficient guarantees of safety and efficiency. Figure 3 shows the products and services involved in this activity. The specific tasks of this activity are described in table 23.

**Design of Control Policies**

In this activity (Figure 33), the set of norms and restrictions needed for applying the normalization dimension is defined. Three different types of standardization are considered:

| Task | Task Description | Roles Involved |
|---|---|---|
| Identify Service Providers | *Functional Dimension Model* diagram is updated by adding the management roles for the OUs that offer services. Additionally, these roles must be added into the *Structural Dimension Model* diagram. | System analyst |
| Detail Services | Every service that the design pattern contributed with must be split into tasks. | System analyst |
| Identify Operative Goals | Functional objectives are split into operative goals, that can be clearly specified. | System analyst |
| Update Functional Dimension Model | After identifying service providers, detailing services and identifying operative goals, *Functional Dimension Model* diagram is updated. | System analyst |
| Detail Environment | System interactions with clients and providers along resources, applications and agents are analyzed in order to determine their correction. New entities like resources and applications are defined to collect all the necessary information of the system. | System analyst |
| Define Domain Ontology | System domain concepts are specified, taking into account the service inputs and outputs, and their relations. Additionally, existing ontologies can be reused in the system. | System analyst |
| Identify Interactions | Relations between OUs are checked in order to create information flows that can facilitate their relationships and information exchange. | System analyst |
| Define Services Interactions | An Interaction entity is defined for every Service, indicating its features and the participants of the interaction. | System analyst |
| Update Environment, Functional and Structural Dimension Models | After detailing environment and identifying interactions, *Environment*, *Functional* and *Structural Dimension Model* diagrams are updated. | System analyst |
| Generate Dynamical Dimension Model Diagrams | After defining service interactions, *Dynamical Dimension Model* diagrams are defined to represent them. | System analyst |

Table 22: *Design of Information-Decision Processes* activity description

standardization of work processes, outputs, and skills.

The *standardization of work processes* implies specifying rules for controlling: (i) invocation and execution order of services; (ii) precedence relationships between tasks; (iii) deadline and activation conditions of services; (iv) and service access to resources or applications.

The *standardization of outputs* implies specifying norms for controlling the service products, based on minimum quality requirements, perceived quality and previous established goals of productivity and performance.

Finally, the *standardization of skills* is integrated in the role concept, which indicates the knowledge and skills required for an agent when playing this role.

As a result (Figure 34), the diagrams of the *Normative Dimension Model* are generated, so the norms needed for controlling the behaviors of the members of the system are detailed. The specific tasks of this activity are described in table 24.

**Design of the Reward System**

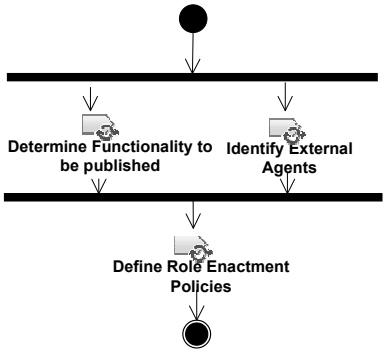



Figure 31: Activity Diagram of *Open System Dynamics* activity.

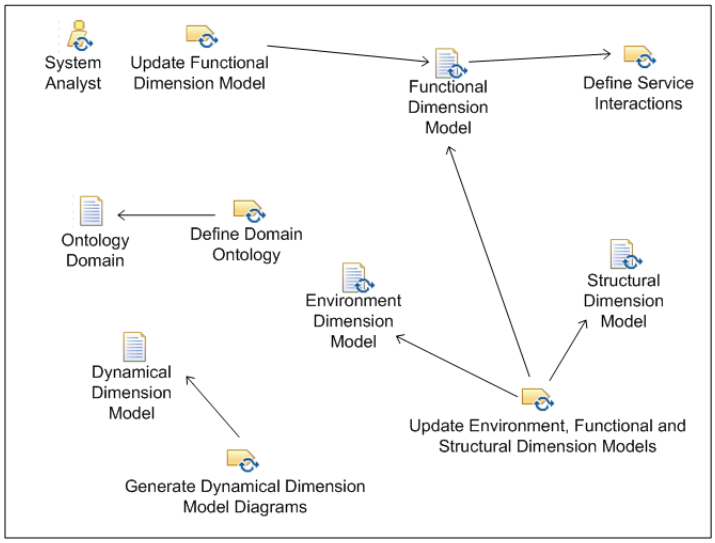



Figure 32: Products of the *Design of Open System Dynamics* activity

This activity defines the reward system needed for allowing the members of the Virtual Organization to work towards the strategy of the organization (Figure 35). Therefore, designer proceeds to:

- Analyze the types of behavior needed to promote:
  - *the willingness to join and remain* inside the system.
  - the *performance dependent on role*, so that the minimal levels of quality and quantity of work to execute are achieved.
  - the *effort on the minimal levels*, defining several measures of performance, efficiency and productivity.
  - the *cooperative behaviors* with the rest of members of the organization.

| Task | Task Description | Roles Involved |
|---|---|---|
| Determine Functionality to be published | Agents must be split into internal and external agents. Every service with the participation of external agents must be published by using a ServicePort entity. | System analyst |
| Identify External Agents | There are two ways for identifying external agents: (i) every external agents has associated an internal agent who represents him, and it is familiar with the interaction protocols; and (ii) to act with the system by means of services and to establish the service management and role enactment rules. | System analyst |
| Define Role Enactment Policies | *AcquireRole* and *LeaveRole* services are split into tasks, taking in account activation conditions, preconditions and restrictions. | System analyst |

Table 23: *Design of Open System Dynamics* activity description

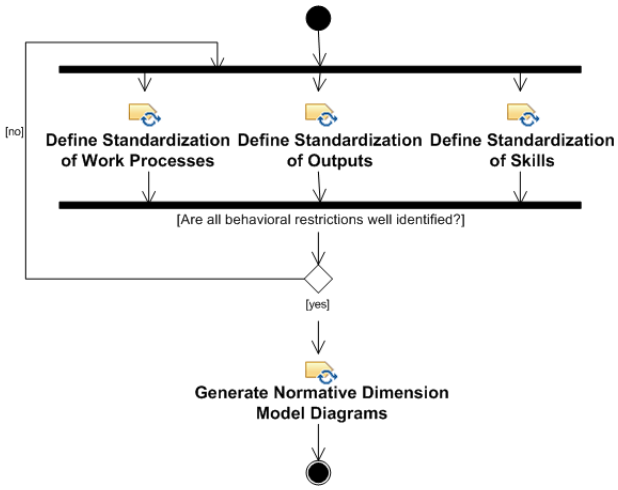



Figure 33: Activity Diagram of *Design of Control Policies* activity.

- Select the type of reward system to be used:
  - *individual rewards*, which establish several measures of behavior of unit members, allowing to promote their efforts on the minimal levels.
  - *group rewards* (competitive or cooperative), that establish several group measures, rewarding unit members based on their specific performance inside the group.
  - *system rewards*, which distribute certain gratifications (ex. permissions, resource accesses) to all members of the system, trying to promote the participation inside the organization.
- Apply the selected reward system in the specific problem domain.

The methodological guideline enables the selection of the type of reward system that should be employed, but it does not detail the concrete mechanisms for implementing this

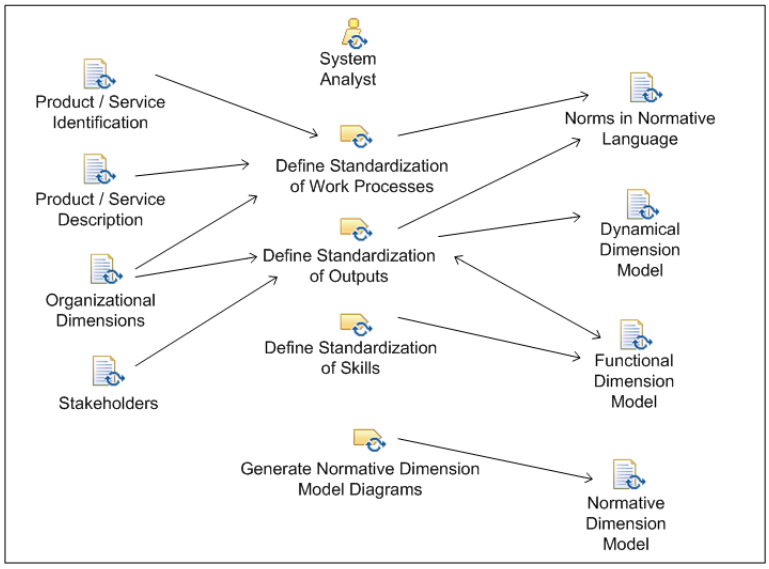



Figure 34: Products of the *Design of Control Policies* activity

system. As a result (Figure 36), the diagrams of the *Normative Dimension Model* are updated, defining new norms or adding sanctions and rewards to the existing norms, according to the selected reward system. The specific tasks of this activity are described in table 25.

### 2.4.3 Work products

The following section describes the products generated by the organization dynamics design phase. Two models are generated: the *Dynamical* and *Normative Dimension Model* diagrams; whereas the *Functional*, *Structural* and *Environment Dimension Model* diagrams are updated. Three structured text documents are generated, regarding the domain ontology, the norms in normative language and the reward system of the organization. Work products are described in table 26. Figure 37 describes their relation with the elements of the GORMAS metamodel.

**Domain Ontology**

To define this structured text document, system domain concepts are specified, taking into account the service inputs and outputs, and their relations. Moreover, if similar ontologies have already been proposed, correspondences between these ontologies and the specific one for the current problem must be established.

GORMAS proposes an organization ontology, that can be employed to describe a system, showed in figure 38. A detailed description of this ontology can be found in [9].

| Task | Task Description | Roles Involved |
|---|---|---|
| Define Standardization of Work Processes | Implies specifying rules for regulating invocation and execution order of services, precedence relationships between tasks, required deadline for execution, access to resources, etc. | System analyst |
| Define Standardization of Outputs | Implies establishing norms to control the results obtained by the service providers, by means of the minimum quality of the produced service or product; and the quality perceived by the stakeholders and to the previous established goals of productivity and performance. | System analyst |
| Define Standardization of Skills | Implies defining the permissions, knowledge and aptitudes than an agent must acquire when taking a role. | System analyst |
| Generate Normative Dimension Model Diagrams | Norms of the Organizational Units are specified, to allow them to establish an execution order between services; to define deadlines or activation/deactivation conditions for services; and to control the access to products and services, using ports. | System analyst |

Table 24: *Design of Control Policies* activity description

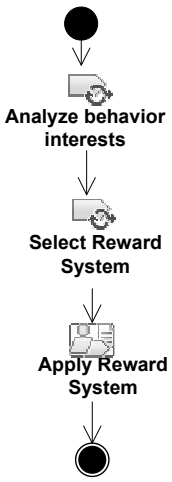



Figure 35: Activity Diagram of *Design of the Reward System* activity.

| Task | Task Description | Roles Involved |
|---|---|---|
| Analyze behavior interests | Analyze the types of behavior needed to promote, like the willingness to join and remain inside the system; the performance dependent or roles; the effort on minimal levels; and the cooperative behaviors in order to create coalitions or groups formed by agents. | System analyst |
| Select Reward System | Select the type of reward system to be used, by describing individual rewards (several measures of behavior of unit member, allowing to promote their efforts on the minimal levels), group rewards (can be cooperative or competitive and are several group measures, rewarding its members based on their specific performance inside the group) and system rewards (gratifications (ex. permissions, resource accesses) to all members of the system, just by belonging to this group, trying to promote the participation inside the organization) | System analyst |
| Apply Reward System | The selected reward system is applied in the specific problem domain. | System analyst |

Table 25: *Design of the Reward System* activity description

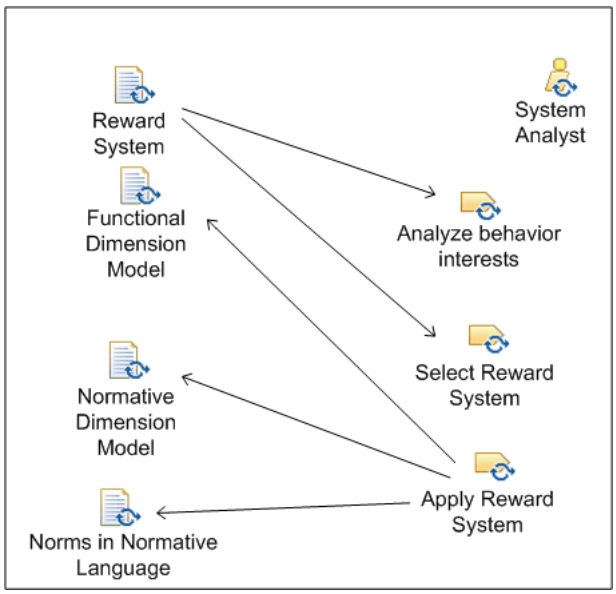



Figure 36: Products of the *Design of Reward System* activity

| **Name** | **Description** | **Work Product Kind** |
|---|---|---|
| Functional Dimension Model | A diagram using the GORMAS graphical notation (based on GOPPR notation) that details the specific functionality of the system, based on services, tasks and goals. | Behavioral |
| Structural Dimension Model | A diagram that uses the GORMAS notation that describes the components of the system and their relationships. It allows defining the static components of the organization. | Structural |
| Environment Dimension Model | A diagram that employs the GORMAS notation that describes the environment elements (resources and applications), along with the agents' behavior. It also allows defining the service ports. | Behavioral |
| Dynamical Dimension Model | A diagram that uses the GORMAS notation that defines the role enactment process, the interactions between agents, as well as the mental states of the entities of the system. | Behavioral |
| Normative Dimension Model | A diagram that uses the GORMAS notation that describes normative restrictions over the behavior of the system entities, including organizational norms and normative goals that agents must follow, including sanctions or rewards. | Behavioral |
| Domain Ontology | A document that specifies the concepts of the organization domain, taking into account the service inputs and outputs. | Structured text |
| Norms in Normative Language | A document that contains a set of norms concerning entities of the system and written in normative language | Structured text |
| Reward System | A document containing a description of the reward system that our organization will apply. | Structured text |

Table 26: Products for *Organization Dynamics Design* phase

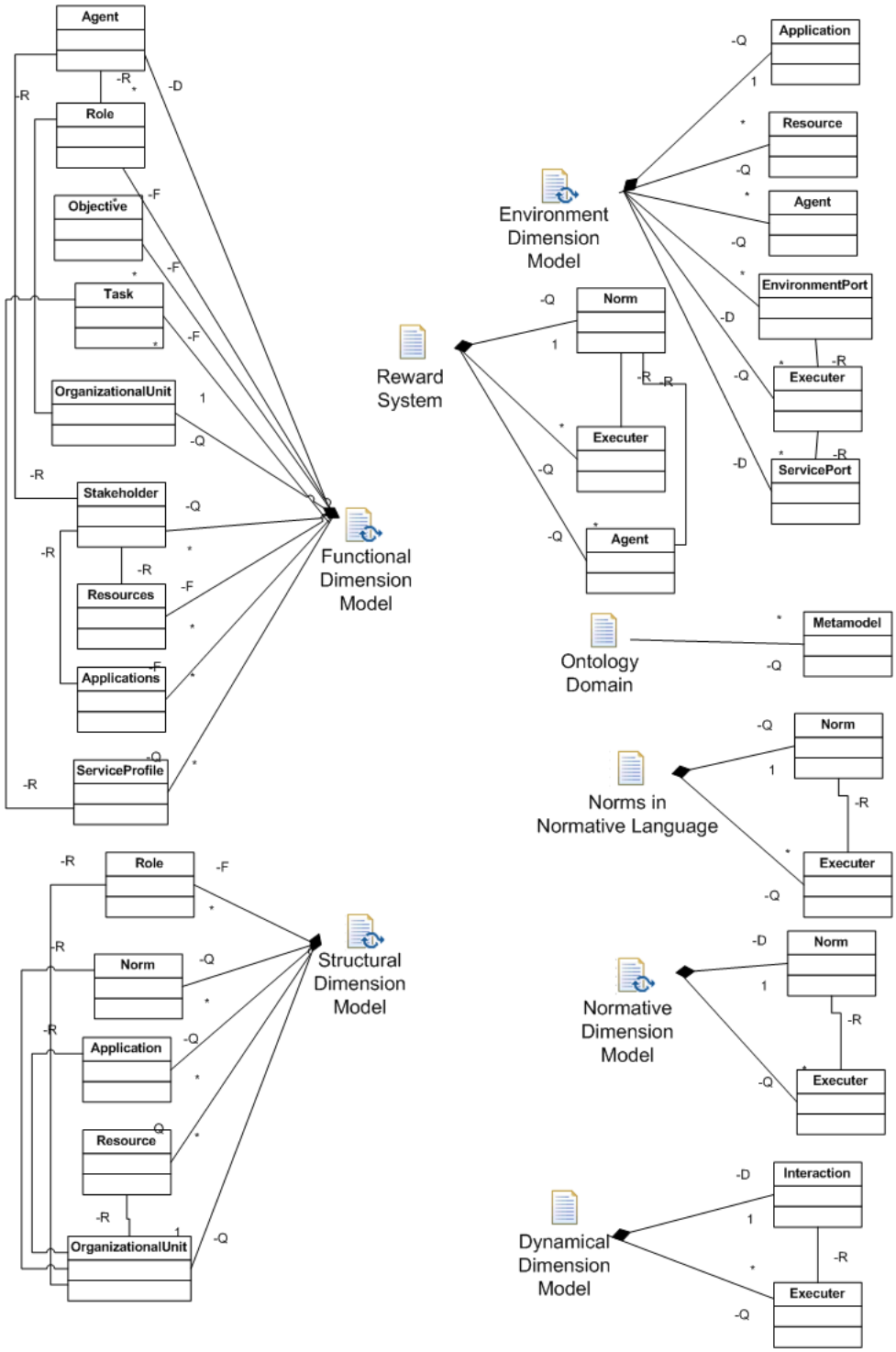



Figure 37: *Organization Dynamics Design* phase. Relations between work products and metamodel elements. **Caption**: D: element introduced for first time; F: element refined; Q: element already defined; R: element related with another element.

| **Entity** | |
|---|---|
| EntityID | String |
| pursues | Goal* |
| is_member | Member* |
| has_resource | Resource* |
| has_portControl | Port* |

| **OrganizationalUnit: Entity** | |
|---|---|
| has_role | Role* |
| has_relationship | Relationship* |
| has_norm | Norm* |
| DomainOntology | Ontology |
| has_member | Member* |
| Type | *FLAT TEAM HIERARCHY* |

| **Role** | |
|---|---|
| RoleID | String |
| Accessibility | *EXTERNAL INTERNAL* |
| Visibility | *PRIVATE PUBLIC* |
| Position | *SUPERVISOR MEMBER SUBORDINATE* |
| Inheritance | Role |
| provides | Service* |
| uses_service | Service* |
| is_affected_by_norm | Norm* |
| is_played_by | Member* |

| **Norm** | |
|---|---|
| NormID | String* |
| norm_defined_in | OrganizationalUnit |
| deonticConcept | *OBLIGED FORBIDDEN PERMITTED* |
| affects | Role |
| issuer | Role* |
| promoter | Role* |
| defender | Role* |
| stateCondition | *BEFORE AFTER BETWEEN* temporalCondition |
| action | *REQUEST SERVE REGISTER* |
| service | Service |
| sanction | Norm |
| reward | Norm |

| **Relationship** | |
|---|---|
| first_role | Role |
| second_role | Role |
| relationship_type | *INFORMATION MONITORING SUPERVISION* |

| **Member** | |
|---|---|
| is_member_of | OrganizationalUnit |
| corresponds_to | Entity |
| plays | Role* |

| **Service** | |
|---|---|
| ServiceID | String |
| is_provided_by | Role* |
| is_used_by | Role* |
| affects_goal | Goal* |
| service_description | OWL-S:Profile |
| service_activity | OWL-S:Process |

| **Resource** | |
|---|---|
| ResourceID | String |
| belongsTo | Entity |
| hasPortAccess | EnvironmentPort* |

| **Port** | |
|---|---|
| controlsPort | Entity |
| is_used_by | Role* |
| has_access_to | Resource Service |

| **EnvironmentPort:Port** | |
|---|---|
| has_access_to | Resource |
| Access_type | *PERCEPT ACT* |

| **ServicePort:Port** | |
|---|---|
| has_access_to | Service |
| Access_type | *REGISTER REQUEST SERVE* |

Figure 38: *Domain Ontology*

| | |
|---|---|
| <norm>::=<deontic> <entity><br><action> [<temporal>]<br>[$IF$ <if_condition>] \| *norm_id* | |
| <ext_norm>::=<norm> [$SANCTION$(<norm>)]<br>[$REWARD$(<norm>)] | |
| <deontic>::=$OBLIGED$ \| $FORBIDDEN$ \|<br>$PERMITTED$ | $\phi : FORBIDDEN\ \alpha \equiv [\alpha]V$<br>$\phi : OBLIGED\ \alpha \equiv [\neg\alpha]V$<br>$\phi : PERMITTED\ \alpha \equiv [\alpha]\neg V$ |
| <entity>::=<agent>: <role> [– <unit>] \|<br><role> [– <unit>] \| <entity_id><br><agent>::=?*variable* \| *agent_id*<br><role>::=?*variable* \| *role_id*<br><unit>::=?*variable* \| *unit_id*<br><entity_id>::=*agent_id* \| *role_id* \| *unit_id* | $\phi : \phi' SANCTION\ \alpha \equiv \phi' \wedge [V]DO(\alpha)$<br>$\phi : \phi' REWARD\ \alpha \equiv \phi' \wedge [\neg V]DO(\alpha)$ |
| <action>::=<functional_action> \|<br><organizational_action> | $\phi : \phi' BEFORE\ \alpha \equiv \phi' \vee DONE(\alpha)$<br>$\phi : \phi' AFTER\ \alpha \equiv [\alpha]\phi'$<br>$\phi : \phi' BETWEEN(\alpha_1, \alpha_2) \equiv [\alpha_1]\phi' \vee DONE(\alpha_2)$ |
| <temporal>::=$BEFORE$ <sit> \| $AFTER$ <sit> \|<br>$BETWEEN$(<sit> , <sit>) | $\phi : \phi' IF\beta \equiv \beta \rightarrow \phi'$ |

Table 27: On the left side, BNF syntax of norms is detailed. On the right side, its semantics expressed by means of dynamic logic is given. $\alpha$ is an action description. $\beta$ is an state description. $V$, $DO(\alpha)$ and $DONE(\alpha)$ are the well-known predicates for representing violation states, an action $\alpha$ that will be done next and an action $\alpha$ that has been performed. Finally, $\phi$ represents a norm.

**Norms in Normative Language**

This structured text document describes the norms that regulate an organization. Norms are employed as mechanisms to limit the autonomy of the agents in complex systems in order to solve complex coordination problems.

In this document, norms are presented using a normative language centered on requesting, serving and registering actions of a concrete service. A detailed description of this normative language can be found in [10]. Tables 27, 28 and 29 show the BNF syntax of the normative language used by the methodology. This is an example of a norm:

```
   PERMITTED HotelProvider REGISTER
HotelReservation PROFILE ProfileRes
ProfileRes= INPUT (hotel:string, company:string,
date:time, rsvNumber:integer, nights:integer)
OUTPUT(ticketRsv:  Reservation, price:  float, IBAN: string)
```

<organizational_action>::=REQUEST <org_service> MESSAGE(<msg_cont>)

<org_service>::=<structural_service> | <dynamic_service> | <informative_service>

<structural_service>::=RegisterNorm | RegisterRole | DeregisterNorm |
DeregisterRole | DeregisterUnit | RegisterUnit

<informative_service>::=InformUnitRoles | InformAgentRoles | InformUnit | InformMembers |
InformRoleProfiles | InformRoleNorms | InformQuantity

<dynamic_service>::=AcquireRole | LeaveRole | Expulse

<functional_action>::=<serv_publication> | <serv_provision> | <serv_usage>

<serv_publication>::=REGISTER *service_name* PROFILE <profile_desc>
[PROCESS<process_desc>]

<service_provision>::=SERVE *service_name* PROCESS <process_desc>
[MESSAGE(<msg_cont>)]

<service_usage>::=REQUEST *service_name* MESSAGE(<msg_cont>)

Table 28: BNF syntax of organizational and functional actions

<profile_desc>::=[INPUT(<param_list>)] [OUTPUT (<param_list>)]
[PRE(<cond_exp>)][POST(<cond_exp>)] | *profile_id*

<process_desc>::=*process_id* | ?*variable* | <action> CONNECTS <process_desc> |
<action> JOIN <process_desc> |
IF <cond_exp> THEN(<process_desc>) [ELSE (<process_desc>)] |
WHILE <cond_exp> DO(<process_desc>)

<msg_cont>::=[SENDER(<entity>)] [RECEIVER (<entity>)]
[PERFORMATIVE (*performative_id*)] CONTENT (<args>)

<action>::=*task_id*(<param_list>) | <service_usage>

<param_list>::=*variable* : *type* [,<param_list>]

Table 29: BNF syntax of service profile and process

**Reward System**

The *Reward System* of an organization describes the general behaviors that should be promoted, specially when using an open system. Rewards can be individual, global or system rewards. GORMAS proposes an analysis about the kind of reward system that best fits the domain, but details of this system are not taken into account. An example of this structured text document is shown in table 30.

**Functional Dimension Model**

The *Functional Dimension Model* diagram was defined on the *Mission Analysis* step (see section 2.1.3) and updated on *Service Analysis* (see section 2.2.3) and *Organizational Design* steps (see section 2.3.3). In the *Organizational Dynamics Design* step, the *Functional Dimension Model* diagram is updated by adding management roles, new services that they can provide and operational objectives. Figure 39 shows the added management roles and

| System Features | Behaviors to promote | Reward system |
|---|---|---|
| Open system<br>There are external agents<br>There are dynamical units<br>*Register / Deregister* unit services were identified | Join and stay | Individual reward<br>System reward |
| High standardization of outputs<br>Conflicts between global goals and individual goals | Effort on minimal levels | Individual reward<br>Group rewards (Competitive) |
| Team type units<br>Group objective | Cooperation | Group rewards (Cooperative) |

Table 30: Example of the document employed as input to help to decide about the *Reward System*

relates them with the services they provide.

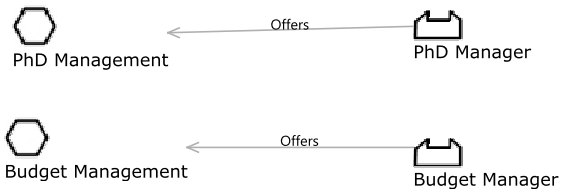



Figure 39: Management roles identified on *Organizational Dynamics Design* phase

**Structural Dimension Model**

The *Structural Dimension Model* was defined on the *Service Analysis* step (see section 2.2.3) and updated on the *Organizational Design* step (see section 2.3.3). In the *Organizational Dynamics Design* step, the *Structural Dimension Model* is updated by adding new roles, resources, applications and their relationships, norms, agents, and by relating OUs with the roles they contain. It is necessary to specify the inheritance of roles. An example of an updated *Structural Dimension Model* is shown in figure 40, which adds a resource (Book), an application (Utilities), two norms (Register and Deregister) and two agents (Director and Student) to the diagram shown in figure 27.

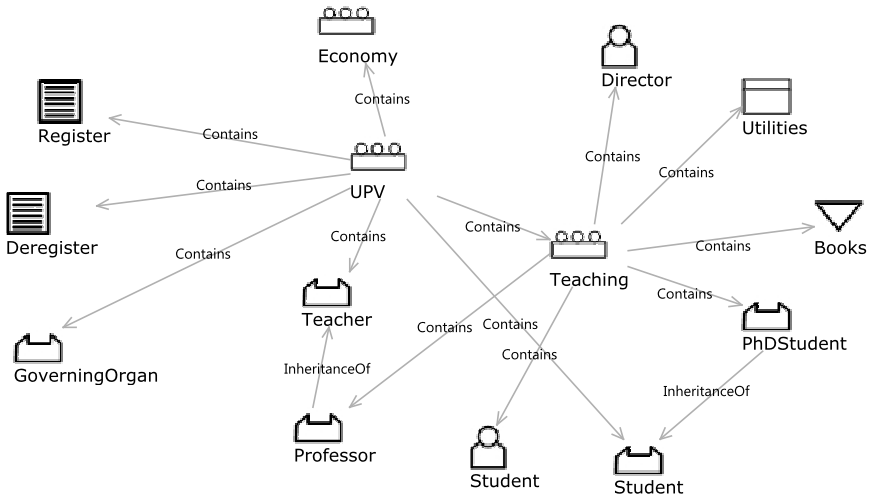



Figure 40: Example of a *Structural Dimension Model* diagram updated on *Organization Dynamic Design* phase

**Environment Dimension Model**

The *Environment Dimension Model* diagram was defined on *Service Analysis* step (see section 2.2.3). In the *Organizational Dynamic Design* step, ports are added, along with their relationships to specify who manages and uses a resource, an application or a service. Figure 41 shows an example of an updated *Environment Dimension Model* diagram. It includes two ports (TotalPort and BookPort); related to two roles (Student and Teacher); a resource (Books) and a application (Calculate totals) contained into an OU (UPV).

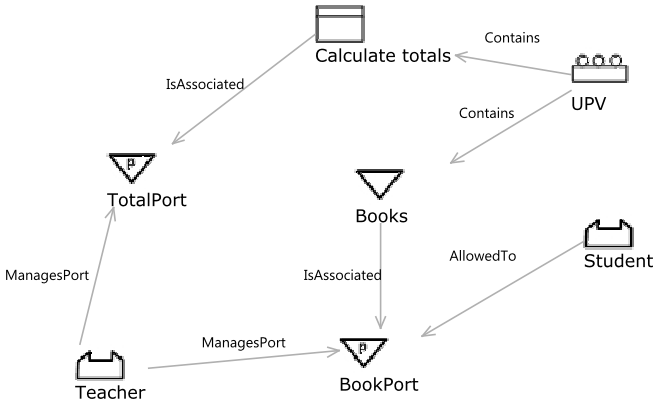



Figure 41: Example of a *Environment Dimension Model* diagram

**Dynamic Dimension Model**

As stated before, the *Dynamic Dimension Model* diagram defines the role enactment process, the interaction between agents, as well as the mental states of the system. In this phase of the methodology, interactions and their participant roles are identified, along with their relationships with objectives. Figures 42 and 43 show an example of a *Dynamic Dimension Model* diagram. Figure 42 shows an interaction between two roles (Teacher and Student), using an interaction entity (Make an exam). Figure 43 shows the different interaction units (performatives) that the interaction runs to execute an interaction. Also, it is depicted a condition that will change the order to execute interaction units if it is fulfilled (student cheating).

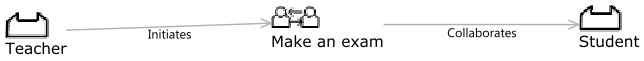



Figure 42: Example of an interaction between two roles in the *Dynamical Dimension Model*.

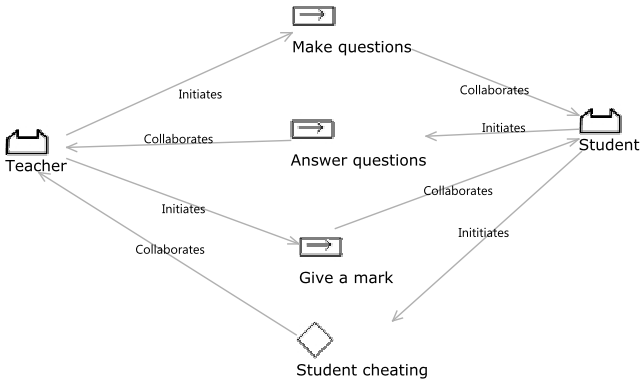



Figure 43: Example of an interaction between two roles in the *Dynamical Dimension Model*, describing the interaction units.

**Normative Dimension Model**

Previously, the *Normative Dimension Model* diagram was defined as a description of normative restrictions over the behavior of the system entities, including organizational norms and normative goals that agents must follow. In this phase of the methodology, norms and their relations with roles, agents, services and objectives are described. Finally, the reward system is specified. Figure 44 shows an example of a *Normative Dimension Model* diagram, depicting two norms (Restriction and Norm) concerning two roles (Student and Teacher).

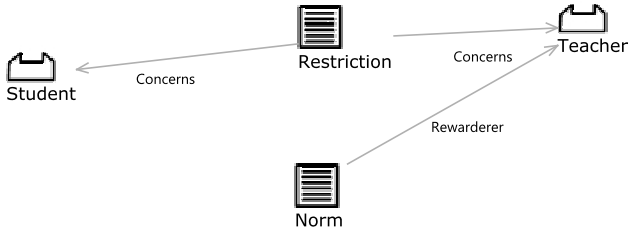
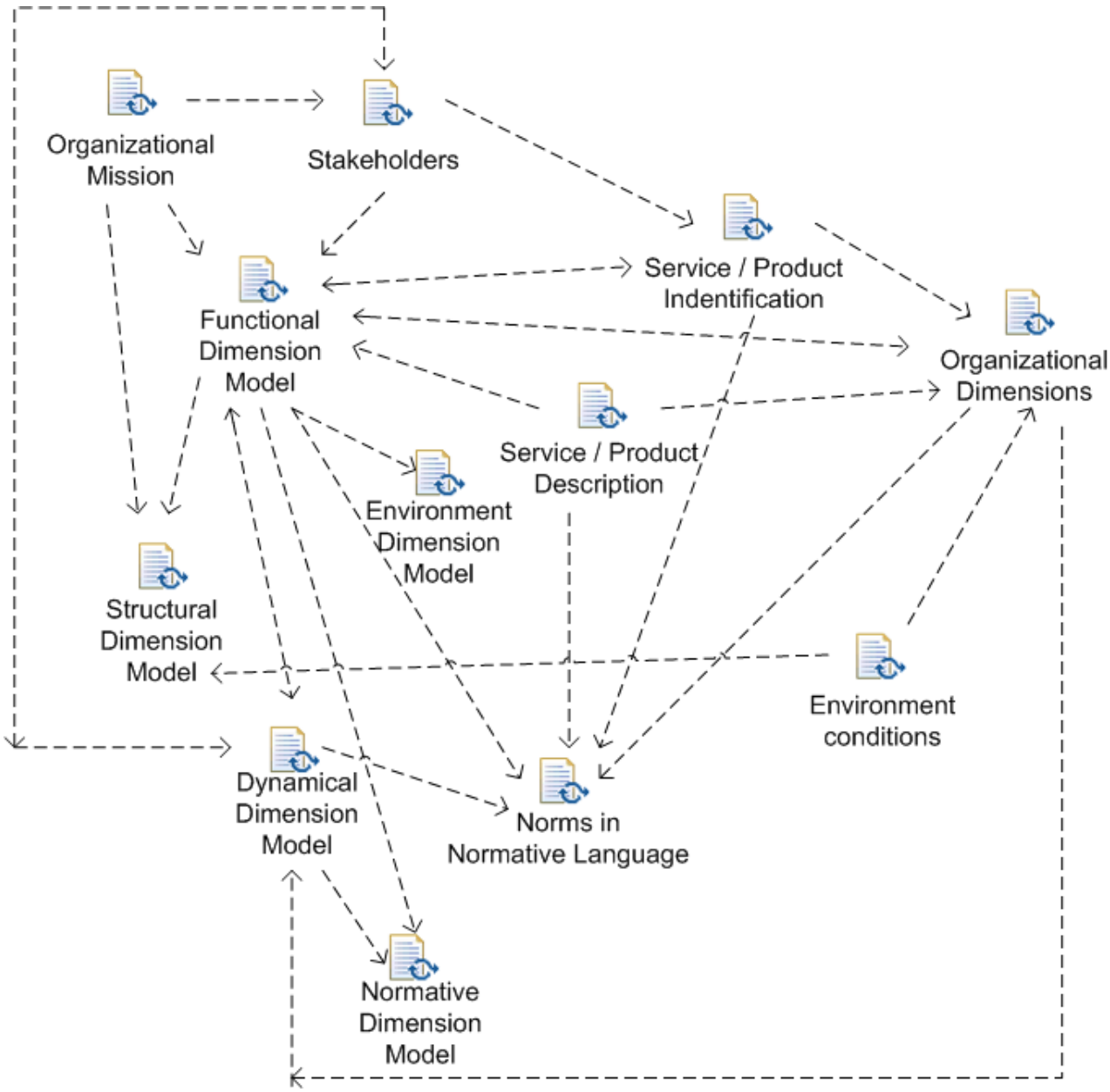



Figure 44: Example of a *Normative Dimension Model* diagram

# 3 Work Product Dependencies

This diagram describes the dependencies among the different work products. For example, the *Organizational Mission* document is used to identify the stakeholders of the organization, that will be described in the *Stakeholders* document. Additionally, the *Organizational Mission* document is used as a guidance to define the *Functional* and *Structural Dimension Model* diagrams.



# References

[1] E. Argente, A. Giret, S. Valero, V. Julian, and V. Botti. Survey of MAS Methods and Platforms focusing on organizational concepts. In *Recent Advances in Artificial Intelligence Research and Development*, volume 113 of *Frontiers in Artificial Intelligence and Applications*, pages 309–316. IOS Press, 2004.

[2] M. Cossentino. From requirements to code with the PASSI methodology. *Agent-oriented methodologies*, pages 79–106, 2005.

[3] A. Omicini. SODA: Societies and Infrastructures in the Analysis and Design of Agent-based Systems. *Agent-Oriented Software Engineering*, 1957:185–193, 2001.

[4] J. Ferber, O. Gutknecht, and F. Michel. From agents to organizations: an organizational view of multi-agent systems. *Lecture Notes in Computer Science*, pages 214–230, 2004.

[5] J. J. Gomez. *Modelado de Sistemas Multi-Agente*. PhD thesis, Universidad Complutense de Madrid, 2002.

[6] A. Giret, V. Julian, M. Rebollo, E. Argente, C. Carrascosa, and V. Botti. An open architecture for service-oriented virtual organizations. In *PROMAS 2009 Post-Proceedings*, pages 1–15. Springer, 2010.

[7] Emilia Garcia, E. Argente, and A. Giret. A modeling tool for service-oriented open multiagent systems modeling tool. In *The 12th International Conference on Principles of Practice in Multi-Agent Systems. PRIMA 2009*, volume 5925 of *LNAI*, pages 345–360. Springer-Verlag, 2009.

[8] S. Kelly, K. Lyytinen, and M. Rossi. MetaEdit+: A fully configurable multi-user and multi-tool CASE and CAME environment. *Lecture Notes in Computer Science*, 1080:1–21, 1996.

[9] N. Criado, E. Argente, V. Julian, and V. Botti. Designing virtual organizations. In *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS2009)*, volume 55 of *Advances in Soft Computing*, pages 440–449, 2009.

[10] N. Criado, V. Julian, V. Botti, and E. Argente. A norm-based organization management system. In *Coordination, Organizations, Institutions, and Norms in Agent Systems*. Springer, 2009.